

AD-A173 699

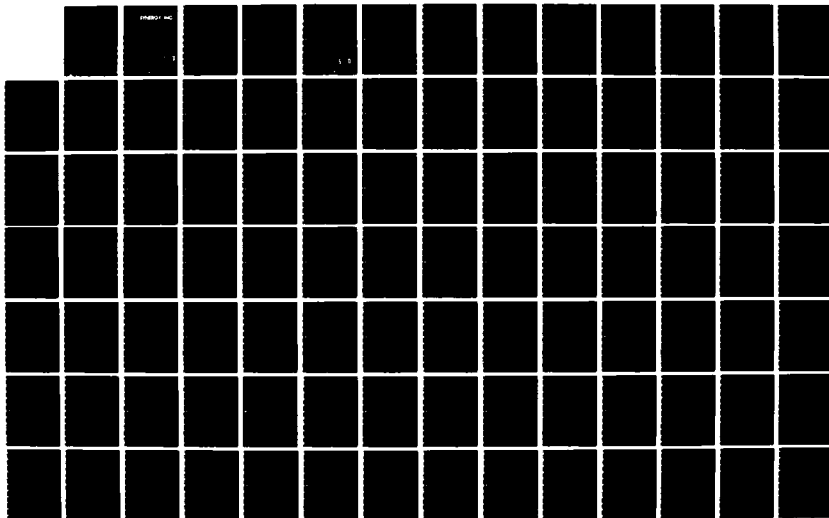
AN EXPERT SYSTEM TO HELP ASSESS TACTICAL AIR READINESS  
AND CAPABILITY PHA (U) SYNERGY INC WASHINGTON DC  
30 MAY 86 F49642-85-D-0035

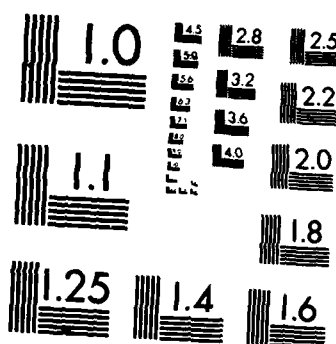
1/2

UNCLASSIFIED

F/G 15/7

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

# SYNERGY, INC.

AD-A173 699

Contract F49642-85-D-0035/5002

DTIC FILE COPY

DTIC  
ELECTE  
NOV 05 1986  
S D E

1824 Biltmore Street, NW  
Washington, DC 20009

This document has been approved  
for public release and sale; its  
distribution is unlimited.

86 10 07

Appendix 2  
The Grammarian's Tool Kit  
Volume I

## APPENDIX 2

— This appendix describes and documents a number of utilities designed to facilitate the creation of a front-end to the expert system which understands natural language. The appendix is divided into three sections: Section I which explains the form of function of the utilities, Section II which includes sample input to the utilities, and Section III which contains two dictionaries and a concordance output by the utilities.

### SECTION I

#### Description of Utilities (Grammarians' Tool Kit)

We built a number of utilities whose purpose is to gain insight into the kinds of questions which might be directed to an expert system to assess tactical air capability. Our initial effort has been directed towards identifying and understanding the individual words and phrases used in the tactical air arena. We have input to our utilities sample questions from a number of sources (see Section II), as well as a lengthy discussion of the Tactical Air Command, which we determined to be a good source of vocabulary.

Our utilities serve the following functions:

- (1) list by frequency or by alphabetical order the vocabulary of the input sample
- (2) show the sentential content in which a particular word appears
- (3) find co-occurrences of certain word pairs
- (4) merge vocabulary lists from different sources
- (5) create a concordance, or list of the sentential contents from different sources for all words in those sources
- (6) create and maintain a dictionary which includes information about the grammatical usage of each word, its meaning, as well as its synonyms, aliases, and abbreviations.

①



APPENDIX 2  
The Grammarian's Tool Kit

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	<i>per</i>
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

F49642-85-D-0035/5002

DTIC  
ELECTE  
NOV 05 1986  
S E D

This document has been approved  
for public release and sale; its  
distribution is unlimited.

The vocabulary builder constructs a list of all the words which appear in the input sample. Along with each word, the program also determines the number of times that the word appears in the input. This occurrence (frequency) count serves to identify words which appear often, and is useful in reasoning about the nature of the input and the best strategy to understand it. There are two ways to display this list: alphabetically, and by frequency. The latter arranges the words in decreasing order of frequency; words having the same frequency are alphabetized within their group.

Another utility, Key Word Out of Context (KWOC), lists all sentences in the input sample which contain a specified word. In the listing, the specified word is centered on the page so that it is easy to compare the contexts of the word in different sentences. This is an essential tool for determining multiple meanings that a word might have, for example:

"How much of this task can we support?"

and

"How many squadrons should we task for DCA CAP?"

When trying to understand English, it is essential to know the part of speech which each word represents. In this example, the way to know that the word "task" is a noun in the first sentence and a verb in the second, is to look at the context. In the first sentence, "task" is immediately preceded by "this" which signals a noun. In the second, "task" is preceded by "we" which requires a verb. It is necessary to specify all of these cases when creating a natural language understanding system.

A third utility facilitates further analysis of words by listing co-occurrences of a pair of words, i.e., all of the sentences in the input sample which contain both words. There are three different ways to specify co-occurrence: ADJACENT, PRIOR, and ANYWHERE. ADJACENT finds those sentences where the first word is immediately followed by the

second word. PRIOR finds those sentences where the first word occurs anywhere before the second word, and ANYWHERE finds all sentences containing both words, regardless of order. Therefore, any sentence found by ADJACENT is also found by PRIOR, and every sentence found by PRIOR is also found by ANYWHERE. For example, suppose we had the following sentences:

1. How many F15's are FMC?
2. How come so many are NMC?
3. Can Bitgurg generate so many sorties?
4. Where can we get so many AGM-45's and how fast can we get them?

(ADJACENT "how" "many") would find sentence 1 only.

(PRIOR "how" "many") would find sentences 1 and 2.

(ANYWHERE "how" "many") would find sentences 1, 2 and 4.

The last three utilities now form part of a functionally cohesive group of utilities, the WORDS utilities. The purpose of the WORDS utilities is to build a lexicon from text files and study the words in context. The WORDS utilities are in the form of five executable files in a directory named WORDS. The utilities operate on some data files which also reside in the WORDS directory. There are six types of files, each with its designated three letter file extension.

Input to the system is a collection of ASCII files. These files must contain text. The period at the end of sentences must be followed by at least either one carriage return or two spaces. The asterisk character \*, has a special meaning -- it marks a comment -- anything on the line appearing after the asterisk is ignored. Carriage returns, line feeds, and tabs are optional, no other control codes are allowed in the file. A file may contain at most 10K bytes. Each of these input files must have the file extension .TXT.

The WORDS program reads a .TXT file and generates a .WOR file with the same filename. The .WOR file contains an alphabetized list of the words appearing in the .TXT file. This is a text file arranged as a sequence of records. Records are delimited by a carriage-return-line-



feed pair. Each record is exactly 30 bytes long. The first 25 bytes contain one word, in upper case, left justified, space padded. The last five bytes contain the number of times that this word appears in the .TXT file right justified, space padded.

The MERGE program combines many .WOR files into one .MRG file. The description of the .MRG file is identical to .WOR files above. The frequency count for each word is the sum of the frequency counts of that word in each of the .WOR files that contain it. This .MRG file is therefore the list of all words, sorted alphabetically.

The SORT-NUM program reads in a .MRG file and generates a .NUM file by permuting the records in the .MRG file. The output is sorted primarily in descending order of frequency, secondarily alphabetically.

The SENTENCE program reads a .TXT file and generates a .SEN file with the same name. This .SEN file contains one line per sentence, question, or exclamation in the .TXT file. Sentence punctuation is omitted in the .SEN file.

The CONCORDANCE program reads in a list of selected words, and a collection of .SEN files and prints out a concordance listing of the selected words. The selected words list (with file extension .SEL), should be a text file with one word per line. The easiest way to generate this .SEL file is to COPY a .MRG file to a .SEL file and use a text editor to remove the unwanted lines. CONCORDANCE ignores anything on the line past the first space.

The PRINTER program is used to print .TXT, .WOR, .MRG, .NUM, .SEN, .SEL, as well as any other text files to the printer. It paginates the output and prints the file name, date, time and page number at the top of each page.

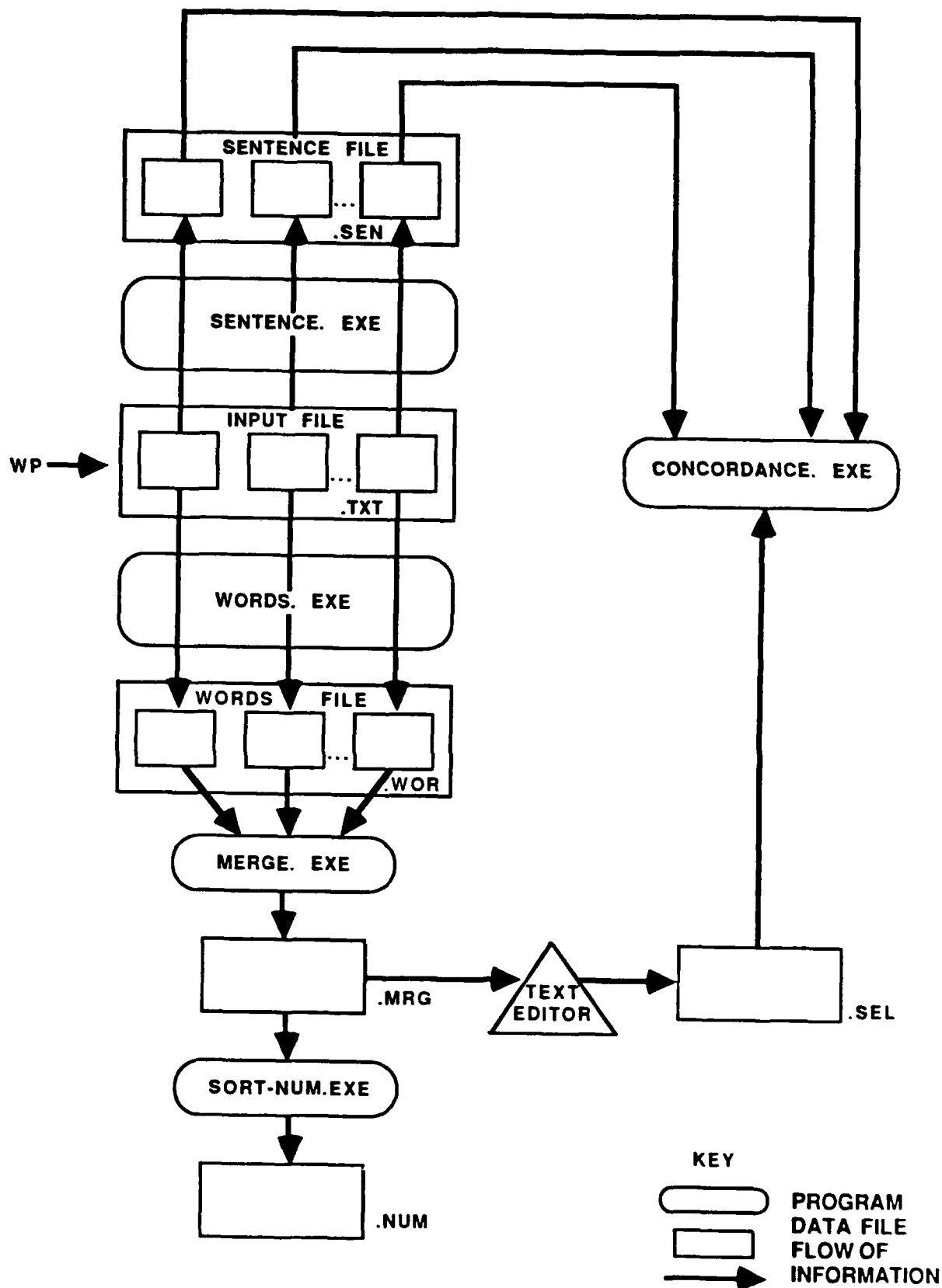


Figure 1: THE WORDS UTILITIES

The system is implemented in BASIC and compiled with the Microsoft Quick BASIC compiler version 1.00. Source resides on .ASC files in the SRC subdirectory of the WORDS directory.

The definition of a word as it pertains to this program is "a sequence of characters delimited by spaces." This definition is perfect for prose which contains nothing but letters (i.e., no digits or punctuation symbols). However, the input to the natural language understanding system will be more complicated. This analysis program does not have a 'deep' understanding of what it is processing, and sometimes does not find the exact location on where one word ends and another begins. This usually happens with punctuation symbols: the hyphen and the slash. For example, one of the 'words' from the questions "How many fighter/recce aircraft are needed ..." is FIGHTER/RECCE. This interpretation is unacceptable in the final system but it is adequate for the level of analysis being performed now.

There is a special kind of a 'word', one which consists only of digits and an optional decimal point. Such a word is a number, represented as "[NUMBER]". Similarly, the pound sign ( ), is represented as "[POUND-SIGN]". We intend to build rules into the final program which will understand special words -- for example 0600Z will be interpreted as a time of day, and 1FS241 will be interpreted as a serial number (with the aid of the context, of course).

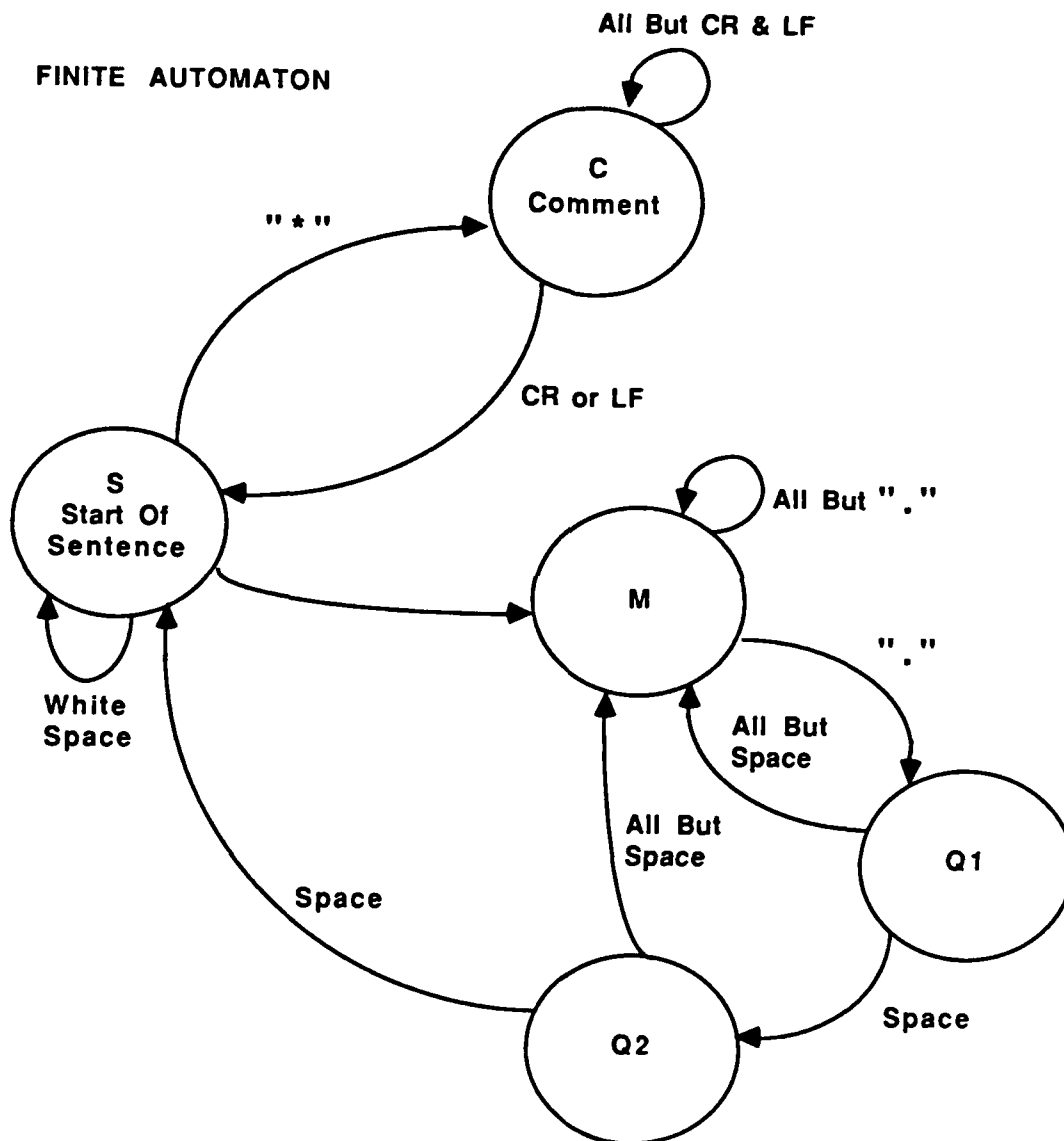


Figure 2: Actual Processing of Natural Language String

The dictionary maintenance utility (DMU) is designed to facilitate entry of lexical, grammatical, and semantic information for each word in the tactical air lexicon. The DMU takes a lexicon generated by the WORDS utilities and creates an entry frame for each word, into which the information can be put. In addition, DMU creates a menu driven format whereby the dictionary writer can:

- E edit the word
- G edit the grammar codes of the word
- D delete the word
- N go to next word alphabetically
- P go to previous word alphabetically
- J jump to a different word
- C create a new word
- T enter aliases or synonyms for the word  
(e.g., "heater" or sidewinder" for AIM-9, "maverick" for AGM-65,  
"partially-mission-capable" for PMC)
- B look at other words to which the word is related in meaning  
(e.g., "F-15" is related to "aircraft" is related to "resource")
- M change default grammar codes for a new word
- S save changes
- Q quit DMU
- ! delete the entire entry for word
- ? help for DMU menu

The DMU is designed to be "user-friendly" so anyone involved in building the expert system can immediately enter information about a word, regardless of that person's linguistic sophistication.

A listing of some of the code  
for the utilities described in Section I

user.l

Functions and documentation to use text analysis functions

```
(defun read1 (filename)
  (gen-sentences filename)
  (terpri)
  (gen2)
  (addemup)
  (makeblist)
  )

;gen-sentences      ;processes sentences for kwoc and co-occurrence
;count-words        ;processes words for alphabetically and
                    ;numerically
```

```
(defun alphabetically nil (show-alphabetically))
  ;sort all words alphabetically, show
  ;word and frequency
```

```
(defun numerically nil (show-by-num))
  ;sort all words using two keys:
  ;primary key is by frequency -- descending
  ;secondary key is alphabetically
```

```
;kwoc    The kwoc function, defined in the file KEYWORD.L
;         accept 1 argument -- a word delimited by blanks
;         [ex. (kwoc " F15's ")]
;         I lists all sentences containing a substring which matches the
;         argument *exactly* (case, white space and all).
```

```
(defun adjacent (word1 word2) (pri (adjacent1 word1 word2)))
  ;show all sentences with word1 immediately
  ;preceeding word2. Ignores case.
  ;usage (adjacent "This" "is")
```

```
(defun prior (word1 word2) (pri (prior1 word1 word2 blist)))
  ;show all sentences containing word1
  ;anywhere before word2
```

```
(defun anywhere (word1 word2) (pri (anywhere1 word1 word2 blist)))
  ;any sentences containing both words
```

```
;load-me.1  
;
```

```
file which loads all files necessary for  
text analysis
```

```
(load "b:msort.1")  
(load "b:co.1")  
(load "b:michael.1")  
(load "b:user.1")  
(load "b:gen-sent.1")  
(load "b:keyword.1")
```



```

; michael.1  functions which do some of the word counting
;
; Stores alist as strings ex. (("THIS" 3) ("IS" 5))

(defun init-alist nil
  (setq alist nil))

(defun count-sentence (sent)
  (mapcar
    #'(lambda (w) (process-word (symbol-name w)))
    sent)
  nil)
; sent is a list of the form
; (IF15'S: 134.3: !FOOBAR!)

(defun process-word (word &aux num)
  (if (numbp word) (setq word "[NUMBER]"))
  (cond ((setq num (cadr (assoc word alist :test 'equal)))
    ; word eg IF15'S:
    ; treat all numbers as 0 here
    ; see if word is in list
    (setq alist (subst (list word (+ num 1))
      (assoc word alist :test 'equal)
      alist))) ; inc count
    (t (setq alist (cons (list word 1) alist)))) ; else add word

(defun numbp (word) (= 0 (length (string-left-trim "1234567890." word))))

; on sort-by-num nil
; msort alist 'cadr-is-lessp))

(defun cadr-is-lessp (pair1 pair2)
  (cond ((= (cadr pair1) (cadr pair2))
    ; if = frequency
    (string-lessp (car pair1) (car pair2))) ; then alphabetically
    (t (> (cadr pair1) (cadr pair2))))
  ; the predicate that sort should use

(defun sort-alphabetically nil
  (msort alist
    #'(lambda (pair1 pair2)
      (string-lessp (car pair1) (car pair2)))))

(defun symbol-name1 (s) (cond ((numberp s) "0") (t (symbol-name s))))

(defun ruff (strin)
  (count-sentence
    (read-from-string strin)))
; rough count from string
; of form "(IF15'S: 13.4: !NOW!)"

(defun show-by-num nil
  (mapcar 'print (sort-by-num))
  nil)

(defun show-alphabetically nil
  (mapcar 'print (sort-alphabetically)) nil)

(defun addemup nil
  (init-alist)
  (mapcar 'ruff slist2))

```

```

(defun msort (l pred &aux l1 l2 l3)
  (cond ((null l) nil)
        ((null (cdr l)) l)
        ((= (length l) 2)
         (cond ((apply pred l) l)
               (t (reverse l))))
        (t (setq l3 (msplit l))
            (setq l1 (msort (car l3) pred) l2 (msort (cadr l3) pred))
            (mmerge l1 l2 pred))))

```

```

(defun msplit (l &aux tmp) ;returns two lists which add up to l each of length
                           ;(length l)/2      Like perfect unshuffle
  (cond ((null l) '(nil nil))
        ((null (cdr l)) (append (list l) '(())))
        (t (setq tmp (msplit (cddr l)))
            (cons (cons (car l) (car tmp))
                  (list (cons (cadr l) (cadr tmp)))))))

```

```

(defun msplit (l)
  ;returns a list of two lists whose append
  ;is a permutation (unshuffle*reverse) of the
  ;original argument

  (prog ((l1 nil) (l2 nil))
    loop
    (if (null l) (return (cons l1 (list l2))))
    (if (null (cdr l)) (return '( , (cons (car l) l1) ,l2)))
    (setq l1 (cons (car l) l1))
    (setq l2 (cons (cadr l) l2))
    (setq l (cddr l))
    (go loop)))

```

```

(defun mmerge (l1 l2 pred)
  ;merge l1 and l2
  (cond ((null l1) l2)
        ((null l2) l1)
        ((apply pred (list (car l1) (car l2)))
         (cons (car l1) (mmerge (cdr l1) l2 pred)))
        (t (cons (car l2) (mmerge l1 (cdr l2) pred)))))

```

```

(defun mmerge (l1 l2 pred)
  ;merge l1 and l2
  (prog ((l nil))
    loop
    (if (null l1) (return (append l2 l)))
    (if (null l2) (return (append l1 l)))
    (if (apply pred (list (car (last l2)) (car (last l1))))
        (progn (setq l (cons (car (last l1)) l))
                (setq l1 (butlast l1)))
        (progn (setq l (cons (car (last l2)) l))
                (setq l2 (butlast l2))))
    (go loop)))

```

;`co.l`

Functions to find co-occurrence of words

```
(defun makeblist nil (setq blist (pairlis slist slist2)) 'Ok)
```

```
(defun adjacent1 (a b) ;returns all sentences with word b  
;immediately following word a.
```

```
;both arguments are strings.
```

```
(adj1 (string-append "!" a "!" b "!") blist))
```

```
(defun adj1 (key l)
```

```
(cond ((null l) nil)
```

```
(t (cond ((string-search* key (cadr l))
```

```
;if matches
```

```
(cons (cadr l)
```

```
(adj1 key (cdr l))))
```

```
;include & recurse
```

```
(t (adj1 key (cdr l))))))
```

```
;else just recurse
```

```
(defun prior1 (a b l)
```

```
;returns all sentences with word b occurring  
;anywhere after word a
```

```
(prior2 (string-append "!" a "!"
```

```
(string-append "!" b "!"
```

```
l))
```

```
(defun prior2 (a b l)
```

```
(cond ((null l) nil)
```

```
(t (append
```

```
(cond
```

```
((and
```

```
(setq temp1 (string-search* a (cadr l))) ;if a is in sent.
```

```
(setq temp2 (string-search* b (cadr l) temp1)))
```

```
;and b after a
```

```
(list (cadr l))))
```

```
;then include
```

```
(prior2 a b (cdr l))))))
```

```
;recurse
```

```
(defun pri (l) (mapcar #'print l) (terpri))
```

```
(defun anywhere1 (a b l)
```

```
(cond ((null l) nil)
```

```
(t (append
```

```
(cond
```

```
((and
```

```
(string-search* a (cadr l))
```

```
;if a is in sent.
```

```
(string-search* b (cadr l)))
```

```
;and b
```

```
(list (cadr l))))
```

```
;then include
```

```
(anywhere1 a b (cdr l))))))
```

```
;recurse
```

;keyword.1 functions to display sentences with a given word

```
(defun kwoc (w)
  (mapcar #'pretty1
    (remove-if
      #'(lambda
          (l)
            (not
              (apply which-string-search
                (list w (string-append " " l " "))))
        slist))
    nil))
```

```
(defun pretty1 (l)
  (setq pos (apply which-string-search (list w (string-append " " l " "))))
  (cond ((> pos 32)
    (setq disp (car (copy-list "("
      (do ((a 0 (+ a 1)) (lcl (min
        (- (length l)
          (- pos 32))
          75)))
        ((= a lcl) nil)
        (setf (aref disp a)
          (aref l (+ a (- pos 32)))))))
    (t
      (setq disp (car (copy-list "("
        (do ((a (- 32 pos) (+ a 1)) (lcl (min
          75
            (+ (length l)
              (- 32 pos))))
          ((= a lcl) nil)
          (setf (aref disp a)
            (aref l (- a (- 32 pos)))))))
      (print disp)
      (do ((a 0 (+ 1 a)))
        ((= a 75) nil)
        (setf (aref disp a) 32)))
```

```
(setq which-string-search 'string-search)
;options: 'string-search case sensitive
;         'string-search* case insensitive
```

```
(defun kwoc1 (w l &aux pos disp)
  (cond ((null l) nil)
    ((setq pos (string-search w (string-append " " (car l) " ")))
      (cond ((> pos 32)
        (setq disp "
          (do ((a 0 (+ a 1)) (lcl (min
            (- (length (car l))
              (- pos 32))
              75)))
            ((= a lcl) nil)
            (setf (aref disp a)
              (aref (car l) (+ a (- pos 32))))))
        (t
          (setf disp "
            (setf (aref disp a)
              (aref (car l) (+ a (- pos 32))))))
```

75

(+ (length (car 1))  
(- 32 pos))))

((= a 1cl) nil)

(setf (aref disp a)

(aref (car 1) (- a (- 32 pos))))))

(print disp)

(do ((a 0 (+ 1 a)))

((= a 75) nil)

(setf (aref disp a) 32))))

(cond (1 (kwocl w (cdr 1))))))

(setq spac "

(defun wai nil

(do ((i 1 (+ 1 i)))

((= i 101) nil)

(do ((a 0 (+ 1 a)))

((= a 75) nil)

(setf (aref spac a) 33))))

;useless function

;outer count

;inner count

```
;gen-sent.l
;                                     program to read in a text file and
;                                     generate slist -- a list of sentences.
;                                     Each sentence is one string, ending with
;                                     a punctuation mark.
;                                     Any text from an asterisk to the end of
;                                     the line is ignored.

;      *** FILES MUST END A ~ (TILDE) ON A NEW LINE ***

; This is a finite automaton with 4 states
; State      Meaning
; -----
; S          Before start of sentence
; M          In a sentence
; Q1         Red a period (46)
; Q2         Red ". "
; C          Reading the rest of a comment

(defun appchar nil (setq s1 (string-append s1 ch)))

(defun list-ascii nil (pprint (list-as 32)))

(defun list-as (c) (cond ((= c 128) nil)
                          (t (cons `(. , (string c)) (list-as (+ c 1))))))

(defun gen-space (&aux a) ;awkward function which returns 32
  (setq a (make-string-input-stream " "))
  (read-char a))

(defun gen-sentences (filename)
  ;outputs - none
  ;side effects: questions, statements, exclamations
  ;               all integer, and together add up to the
  ;               number of sentences red.
  ;               slist -- a list of sentences of the form
  ;               ("The cat is in the hat" "Where's the cat?")
  (setq slist nil) ;init output
  (setq questions 0)
  (setq statements 0)
  (setq exclamations 0)
  (prog (stream state s1 ch)
    (setq stream (open filename)) ;open file
    (setq state 'S)
    (setq s1 "")
    loop
    (setq ch (read-char stream))
    (if (= ch 126) (close stream)) ;eof mark
    (if (= ch 126) (return nil)) ;eof mark
    (if (= ch 42) (setq state 'C)) ;comment -- ignore
    (if (and (= ch 10) (equal state 'C))
        (setq state 'S) ;end of comment
    (if (and (= ch 10) (equal state 'Q1))
        (setq state 'Q2) ;convert "."+newline to ". "
    (if (= ch 10) (setq ch (gen-space))) ;LF --> SPACE
    (if (= ch 13) (go loop)) ;ignore CR
    (if (= ch 39) (go loop)) ;throw away "'"
    (if (= ch 44) (go loop)) ;throw away ","
    (if (= ch 35) ;#
        (progn nil
              (setq s1 (string-append s1 "[POUND-SIGN]"))
              (go loop)))
```

```

(cond ((equal ch 32) (go loop))
      (t (setq s1 (string ch))
          (setq state 'M)
          (go loop))))

((equal state 'M)
 (cond ((or (= ch 33) (= ch 63)) ; "?" or "!"
        (tally-sentence-type (string ch))
        (setq slist (append slist (list s1)))
        (setq state 'S)
        (go loop))
        ((= ch 46) ; "."
        (setq state 'Q1)
        (go loop))
        (t
         (appchar)
         (go loop))))

((equal state 'Q1)
 (cond ((= ch 32) ; " "
        (setq state 'Q2)
        (go loop))
        (t
         (setq s1 (string-append s1 ".")) ;from before
         (appchar)
         (setq state 'M)
         (go loop))))

((equal state 'Q2)
 (cond ((= ch 32)
        (tally-sentence-type ".")
        (setq slist (append slist (list s1)))
        (setq state 'S)
        (go loop))
        (t
         (setq s1 (string-append s1 ". ")) ;from before
         (appchar)
         (setq state 'M)
         (go loop))))

((equal state 'C) (go loop)))

(mapcar #'print slist)
nil)

```

```

(defun tally-sentence-type (type)
  (cond ((equal type ".") (setq statements (+ statements 1)))
        ((equal type "!") (setq exclamations (+ exclamations 1)))
        ((equal type "?") (setq questions (+ questions 1)))))

```

```

(defun gen2 nil (setq slist2 (gen2r slist)) (mapcar #'print slist2) nil)

```

```

(defun gen2r (l) ;recursive
  (cond ((car l)
        (prog (is os i ch) ;is : input string
              ;os : output string
              ;i : index of is
              ;ch : current char
              (setq is (car l))
              (setq os "(!")
              (setq i 0)
              loop)
        (t)

```

```

(cond ((= ch 32) (setq os (string-append os "!"))
      (return (cons os (gen2r (cdr 1))))))
(setq ch (aref is i))
(if (and (> ch 96) (< ch 123))
    (setq ch (- ch 32))) ;UPPERFY
(cond ((= ch 32) ;white space
      (setq os (string-append os "! !")) ;word delimiter
      (do ((a i (+ a 1)))
          ((not (= (aref is a) 32)) (setq i a)))
      ;skip white space
      (go loop)))

(t ;a word
  (setq os (string-append os (string ch)))
  (setq i (+ 1 i))
  (go loop))))))

```

```

(defun count-words (filename)
  ;outputs - none
  ;side effects: questions, statements, exclamations
  ;               all integer, and together add up to the
  ;               number of sentences read.
  ;               alist -- a list of pairs, each of the form
  ;               ("WORD" 3). Ordering is arbitrary

  (setq alist nil)
  (setq questions 0)
  (setq statements 0)
  (setq exclamations 0)
  (prog (stream state s1 ch)
    (setq stream (open filename)) ;open file
    (setq state 'S)
    (setq s1 "")
    loop
    (setq ch (read-char stream))
    (if (= ch 126) (close stream)) ;eof mark
    (if (= ch 126) (return nil)) ;eof mark
    (if (= ch 42) (setq state 'C)) ;comment -- ignore
    (if (and (= ch 10) (equal state 'C))
        (setq state 'S)) ;end of comment
    (if (and (= ch 10) (equal state 'Q1))
        (setq state 'Q2)) ;convert "."+newline to ". "
    (if (= ch 10) (setq ch (gen-space))) ;LF --> SPACE
    (if (= ch 13) (go loop)) ;ignore CR
    ; (if (= ch 39) (go loop)) ;throw away '"'
    (if (= ch 44) (go loop)) ;throw away ","
    (if (= ch 35) ;#
        (progn nil
              (setq s1 (string-append s1 "[FOUND-SIGN]"))
              (go loop)))

    (cond ((equal state 'S)
          (cond ((equal ch 32) (go loop))
                (t (setq s1 (string ch))
                   (setq state 'M)
                   (go loop))))

          ((equal state 'M)
           (cond ((or (= ch 33) (= ch 63)) ;"?" or "!"
                 (tally-sentence-type (string ch))
                 (setq sent2 (car (gen2r (list s1))))
                 (ruff sent2)
                 (setq state 'S))

```



```

    (setq state 'Q1)
    (go loop))
  (t
    (appchar)
    (go loop))))

((equal state 'Q1)
 (cond ((= ch 32) ;" "
        (setq state 'Q2)
        (go loop))
       (t
        (setq s1 (string-append s1 ". ")) ;from before
        (appchar)
        (setq state 'M)
        (go loop))))

((equal state 'Q2)
 (cond ((= ch 32)
        (tally-sentence-type ".")
        (setq sent2 (car (gen2r (list s1))))
        (ruff sent2)
        (setq state 'S)
        (go loop))
       (t
        (setq s1 (string-append s1 ". ")) ;from before
        (appchar)
        (setq state 'M)
        (go loop))))

((equal state 'C) (go loop)))

```

nil)

```

(defun write-list (l filename)
  (setq stream (open filename :direction :output))
  (mapcar #'(lambda (a) (format stream "~A,~D ~%" (car a) (cadr a))) l)
  (close stream)
  "File written.")

```

```

1  WORDS          READ .TXT FILE, GENERATE .WOR FILE
2  4/14/86
3
4
5  INPUT FILE FORMAT:      PLAIN TEXT FILE
6
7  OUTPUT FILE FORMAT:     TEXTFILE OF LINES.  EACH LINE IS 30 BYTES.
8                           BYTES 1-25 ARE THE WORD.  BYTES 26-30 ARE
9                           THE FREQUENCY.  FREQUENCY IS RIGHT JUSTIFIED.
10
11
100 DEFINT A-Z:ON ERROR GOTO 12000:COLOR 15,0
120 DIM HASH[1000,8]'      HASH TABLE:  10 DEEP, index into W$[] and W[]
122                           HASHLENGTH=1000:HASHDEPTH=8
130 DIM W$[3000]'          3000 WORDS
140 DIM W[3000]'           frequency count
145 DIM S[3000]'           pointers into W$[], sorted
147 DIM TEMP[3000]
150 W$[0]="":W=0'          W=# of words so far
160 DIM SL[13],SR[13]'     'recursion stack
170 WHICH.FUNCTION=1'      '1 for alphabetically, 2 for frequency
500 '***** MAIN PROGRAM *****
505 CLS:PRINT "WORDS          FIND WORDS IN TEXTFILE
510 GOSUB 2000'            'read words
520 GOSUB 7000'            'generate list
530 GOSUB 9000'            'write list
540 END
550 '***** read input *****
2010 PRINT:INPUT "enter input filename [DO NOT ENTER EXTENSION.  .TXT IS ASSUMED
] ",I$
2015 IF INSTR(I$,".") THEN BEEP:GOTO 2010
2020 OPEN "I",1,I$+".TXT":GOTO 2050
2030 BEEP:PRINT "ERROR.  Try again.":GOTO 2010
2050
2055 WORD$=""
2060 INWORD=0
2070 WHILE A$(">"~" AND NOT(EOF(1))
2080 A$=INPUT$(1,1):IF A$(">"CHR$(10) AND A$(">"~" THEN PRINT A$;
2085 IF A$="~" THEN LINE INPUT #1,A$:GOTO 2300 'Comment
2090 IF INSTR(" ,.;?/@#&()" +CHR$(10)+CHR$(13),A$) THEN 2150
2100 'not a space
2110 IF INWORD THEN WORD$=WORD$+A$ ELSE WORD$=A$:INWORD=-1
2120 GOTO 2300
2150 'space or delimiter
2160 IF NOT INWORD THEN 2300 ELSE GOSUB 3000:WORD$="":INWORD=0
2300 WEND
2330 CLOSE 1
2340 PRINT
2980 PRINT USING "##### unique words";W
2990 PRINT USING "##### total words";TOTAL.WORDS
2999 RETURN
3000 '***** ADD A WORD *****
3010 'INPUT WORD$          WORD TO ADD
3020 'OUTPUT updates HASH$[], W$[], W[]
3022 GOSUB 6500
3025 TOTAL.WORDS=TOTAL.WORDS+1
3030 HASH$=WORD$:GOSUB 4000
3040 I=0:WHILE W$[HASH[HASH,I]](">"WORD$ AND HASH[HASH,I](">"0:I=I+1:WEND
3050 IF HASH[HASH,I]=0 THEN 3090
3060 W[HASH[HASH,I]]=W[HASH[HASH,I]]+1

```

```

3070 RETURN
3080 W=W+1:W$(W)=WORD$:W(W)=1:HASH(HASH,I)=W
3090 RETURN
4000 '***** COMPUTE HASH FUNCTION *****
4010 'INPUT HASH$ string to hash
4020 'OUTPUT HASH integer 0..HASHLENGTH-1
4030 HASH=0:FOR I = 1 TO LEN(HASH$):HASH=HASH+ASC(MID$(HASH$,I,1)):NEXT
4040 HASH=HASH MOD HASHLENGTH
4050 RETURN
6500 '***** GIVEN WORD$, CHECK IF NUM *****
6510 TM = 1:BUSTED=0:WHILE BUSTED=0
6520 IF INSTR("1234567890.",MID$(WORD$,TM,1))=0 THEN BUSTED=-1:GOTO 6540
6530 IF TM=LEN(WORD$) THEN BUSTED=-2 ELSE TM=TM+1
6540 WEND
6550 IF BUSTED=-2 THEN WORD$="[NUMBER]" ELSE UP$=WORD$:GOSUB 17000:WORD$=UP$
6570 RETURN
7000 '***** GENERATE SORTED LIST *****
7010 'OUTPUT S[1..W]
7020 FOR I = 1 TO W:S[I]=I:NEXT:PRINT
7030 SP=0:SL[0]=1:SR[0]=W:GOSUB 8000 'SP=STACK POINTER
7040 LOCATE ,1,1:PRINT " ":RETURN
8000 '***** RECURSIVELY SORT S[SL[SP]..SR[SP]] *****
8002 LOCATE ,SP+1,0:PRINT ". ";
8005 'PRINT SPACE$(SP)"enter SORT["SL[SP];SR[SP]"]";:Q=SP:GOSUB 50000
8010 IF SR[SP]=SL[SP] THEN RETURN
8020 IF SR[SP]>SL[SP]+1 THEN 8050 ELSE P1=SL[SP]:P2=SR[SP]:
ON WHICH.FUNCTION GOSUB 10000,11000
8030 IF NOT RESULT THEN SWAP S[P1],S[P2]:RETURN ELSE RETURN
8050 SP=SP+1:SL[SP]=SL[SP-1]:SR[SP]=(SR[SP-1]+SL[SP-1])/2:GOSUB 8000
8052 LOCATE ,SP+1,0:PRINT ". ";
8060 SL[SP]=SR[SP]+1:SR[SP]=SR[SP-1]:GOSUB 8000:SP=SP-1
8062 LOCATE ,SP+1,0:PRINT ". ";
8070 P1=SL[SP]:P2=1+(SL[SP]+SR[SP])/2:MIDPOINT=P2-1
8080 FOR T = SL[SP] TO SR[SP]
8085 GOSUB 10000
8090 IF RESULT THEN TEMP=S[P1]:P1=P1+1 ELSE TEMP=S[P2]:P2=P2+1
8100 TEMP[T]=TEMP
8110 IF P1>MIDPOINT THEN FOR II = T+1 TO SR[SP]:TEMP[II]=S[P2]:
P2=P2+1:NEXT:T=SR[SP]:GOTO 8300
8120 IF P2>SR[SP] THEN FOR II = T+1 TO SR[SP]:TEMP[II]=S[P1]:
P1=P1+1:NEXT:T=SR[SP]:GOTO 8300
8300 NEXT
8310 FOR T = SL[SP] TO SR[SP]:S[T]=TEMP[T]:NEXT
8410 RETURN
9000 '***** WRITE SORTED LIST *****
9010 OPEN "O",2,1$+".WOR"
9050 FOR I = 1 TO W
9062 PRINT #2,W$(S[I]) TAB(26);:PRINT #2,USING "#####";W(S[I])
9080 NEXT
9085 CLOSE 2
9090 RETURN
10000 '***** ALPHABETICALLY *****
10010 'INPUT: P1, P2 pointers into S[]
10020 'OUTPUT: RESULT -- -1 if P1 should come before P2
10030 RESULT=W$(S[P1])<W$(S[P2]):RETURN
11000 '***** BY FREQUENCY *****
11010 'INPUT: P1, P2 pointers into S[]
11020 'OUTPUT: RESULT -- -1 if P1 should come before P2
11030 IF W(S[P1])>W(S[P2]) THEN RESULT=-1 ELSE IF W(S[P1])=W(S[P2]) THEN
RESULT=W$(S[P1])<W$(S[P2]) ELSE RESULT=0

```

```
11040 RETURN
12000 '***** ERROR TRAPPING *****
12010 IF ERL=2020 THEN RESUME 2030
12900 ON ERROR GOTO 0
17000 '***** UPPERFY      UP$ *****
17010 'INPUT AND OUTPUT:  UP$
17020 FOR U = 1 TO LEN(UP$):UP=ASC(MID$(UP$,U,1)):IF UP>96 AND UP<123
      THEN MID$(UP$,U,1)=CHR$(UP-32)
17030 NEXT:RETURN
```

```

1 .
2 .
3 .
4 .
5 .
MERGE                                Program to merge .WOR files into a .MRG file

100 DEFINT A-Z:ON ERROR GOTO 12000
110 DIM F$(100)                      'up to 100 input file names
500 COLOR 7,0:CLS:PRINT "MERGE      Program to merge .WOR files into a .MRG
file
510 PRINT :PRINT
520 INPUT "ENTER FILE TO CONTAIN THE MERGED DATA [OMIT THE .MRG FILE EXT.] ",M$
530 IF INSTR(M$,".")>0 THEN BEEP:GOTO 520 ELSE OM$=M$:M$=M$+".MRG"
540 OPEN "I",1,M$
550 CLOSE 1
560 PRINT M$ " ALREADY EXISTS. ":INPUT "DO YOU WANT TO MERGE FILES INTO IT? ",Q$
570 Q$=CHR$(ASC(Q$+" ") AND 223)
580 IF Q$="N" THEN OPEN "R",1,OM$+".BAK":CLOSE 1:KILL OM$+".BAK":
NAME M$ AS OM$+".BAK":OPEN "O",1,M$:CLOSE 1:GOTO 700
590 IF Q$<>"Y" THEN BEEP:GOTO 560
700 'NO PROBLEM
705 OPEN "R",1,M$:CLOSE 1:PRINT
710 INPUT "ENTER FILE NAME [OMIT .WOR EXTENSION] TO READ, [RETURN TO START] ",F$
720 IF F$="" THEN 1000
730 OPEN "I",1,F$+".WOR":CLOSE 1:F=F+1:F$(F)=F$:GOTO 710
740 BEEP:PRINT "FILE NOT FOUND!":GOTO 710
1000 'PROCEED WITH MERGE
1010 OPEN "R",1,"%MERGE.TMP":CLOSE 1:KILL "%MERGE.TMP"
1020 FOR I = 1 TO F
1030 PRINT F$(I)
1040 NAME M$ AS "%MERGE.TMP"
1050 OPEN "I",1,"%MERGE.TMP":GOSUB 3000
1060 OPEN "I",2,F$(I)+ ".WOR":GOSUB 4000
1070 OPEN "O",3,M$
1080 AW$=LEFT$(A$,25):BW$=LEFT$(B$,25)
1090 IF AW$=BW$ THEN IF A$="}" THEN 1490 ELSE PRINT #3,AW$;TAB(26);:
PRINT #3,USING "#####";VAL(MID$(B$,26))+VAL(MID$(A$,26)):GOSUB 3000:
GOSUB 4000:GOTO 1080
1100 IF AW$<BW$ THEN PRINT #3,A$:GOSUB 3000:GOTO 1080
1110 IF BW$<AW$ THEN PRINT #3,B$:GOSUB 4000:GOTO 1080
1490 CLOSE 1,2,3
1500 KILL "%MERGE.TMP"
1510 NEXT
1590 END
3000 'NEW LINE
3010 LINE INPUT #1,A$
3020 RETURN
4000 'NEW LINE
4010 LINE INPUT #2,B$
4020 RETURN
12000 '***** ERROR TRAPPING *****
12010 IF ERR=53 AND ERL=540 THEN RESUME 700
12020 IF ERR=53 AND ERL=730 THEN RESUME 740
12030 IF ERR=62 THEN IF ERL=3010 THEN A$="}":RESUME 3020
12040 IF ERR=62 THEN IF ERL=4010 THEN B$="}":RESUME 4020
12999 ON ERROR GOTO 0:STOP

```

```

1 .
2 .      SORT-NUM      READ  A .MRG FILE SORTED ALPHABETICALL,
3 .      WRITE A .NUM FILE SORTED BY FREQUENCY
4 .
5 .
100 DEFINT A-Z
105 DIM N(20000)
110 CLS:PRINT "SORT-NUM      SORT BY FREQUENCY"
120 PRINT :INPUT "ENTER INPUT FILE [OMIT THE .MRG EXTENSION] ",I$
125 IF INSTR(I$,".")>0 THEN BEEP:GOTO 120
130 PRINT :INPUT "ENTER OUTPUT FILE [OMIT THE .NUM EXTENSION] ",O$
135 IF INSTR(O$,".")>0 THEN BEEP:GOTO 130
137 PRINT
140 OPEN "I",1,I$+".MRG"
150 WHILE NOT EOF(1)
160     LINE INPUT #1,A$:N=VAL(MID$(A$,26)):N(N)=N(N)+1:IF N>MAX THEN MAX=N
170 WEND
180 CLOSE 1
185 OPEN "O",2,O$+".NUM"
190 FOR I = MAX TO 1 STEP -1
200     IF N(I)=0 THEN 400
210     OPEN "I",1,I$+".MRG"
215     LOCATE ,1:PRINT I;
220     FOR J = 1 TO N(I)
230         LINE INPUT #1,A$:IF VAL(MID$(A$,26))=I THEN PRINT #2,A$
        ELSE GOTO 230
    NEXT
    CLOSE 1
NEXT
CLOSE 2
420 END

```

```

1 .
2 .
3 SENTENCE READ .TXT FILE, GENERATE .SEN FILE
4 .
5 4/14/86
6 .
100 DEFINT A-Z:ON ERROR GOTO 12000:COLOR 15,0
500 '***** MAIN PROGRAM *****
505 CLS:PRINT "SENTENCE CREATE .SEN FILE FROM .TXT FILE
560 GOSUB 2000:CLOSE
790 GOTO 560
2000 '***** find sentences *****
2010 PRINT:INPUT "ENTER FILE NAME [OMIT .TXT FILE EXTENSION] [RETURN TO END] ",I$
IF I$="" THEN END
2015 IF INSTR(I$,".") THEN BEEP:GOTO 2010
2020 OPEN "i",1,I$+".TXT":GOTO 2050
2030 BEEP:PRINT "ERROR. Try again.":GOTO 2010
2050 OPEN "D",2,I$+".SEN"
2055 STATE$="S"
2060 A$=" ":S$=""
2070 WHILE A$<>"~" AND NOT(EOF(1))
2075 IF INKEY$=CHR$(3) THEN STOP 'CONTROL C
2080 A$=INPUT$(1,1):A=ASC(A$) 'loop
2090 IF A<>10 AND A$<>"*" THEN PRINT A$;
2095 IF A=9 THEN A=32:A$=" "
2100 IF A$="*" THEN LINE INPUT #1,A$:STATE$="S":GOTO 2900
2110 IF A=10 THEN A=13 'convert CR to LF
2120 IF A=13 AND STATE$="Q1" THEN STATE$="Q2"
2130 IF A=13 THEN A=32:A$=" "
2140 IF INSTR(",','",A$) THEN 2900 'throw away garbage
2150 IF A$="#" THEN S$=S$+"[POUND-SIGN]":IF STATE$="Q1" OR STATE$="Q2"
THEN STATE$="M":GOTO 2900
2160 IF STATE$="S" THEN IF A=9 OR A=32 THEN 2900 ELSE STATE$="M":
S$=A$:GOTO 2900
2170 IF STATE$="M" THEN IF A$="?" THEN QUESTIONS=QUESTIONS+1:ELSE
IF A$="!" THEN EXCLAMATIONS=EXCLAMATIONS+1
2180 IF STATE$="M" AND INSTR("?!",A$)>0 THEN S=S+1:PRINT #2,S$:
STATE$="S":GOTO 2900
2190 IF STATE$="M" THEN IF A$="." THEN STATE$="Q1":GOTO 2900 ELSE S$=S$+A$:
GOTO 2900
2200 IF STATE$="Q1" THEN IF A$=" " THEN STATE$="Q2":GOTO 2900 ELSE
S$=S$+"."+A$:STATE$="M":GOTO 2900
2210 IF STATE$="Q2" THEN IF A$=" " THEN STATEMENTS=STATEMENTS+1:S=S+1:
PRINT #2,S$:STATE$="S":GOTO 2900 ELSE STATE$="M":
S$=S$+"."+A$:GOTO 2900
2220 PRINT "ERROR!":STOP
2900 'end of loop
2950 WEND
2952 CLOSE 1
2955 PRINT
2960 PRINT USING "#### questions";QUESTIONS
2970 PRINT USING "#### statements";STATEMENTS
2975 PRINT USING "#### exclamations";EXCLAMATIONS
2980 RETURN
12000 '***** ERROR TRAPPING *****
12010 IF ERL=2020 THEN RESUME 2030
12900 ON ERROR GOTO 0
16540 RETURN
17000 '***** UPPERFY UP$ *****
17010 INPUT AND OUTPUT: UP$

```

```
17020 FOR U = 1 TO LEN(UP$):UP=ASC(MID$(UP$,U,1)):IF UP>96 AND UP<123  
    THEN MID$(UP$,U,1)=CHR$(UP-32)
```

```
18030 NEXT:RETURN
```

```
19000 FOR KKK = SL(Q) TO SR(Q):PRINT W$(S[KKK])" "":NEXT:PRINT :RETURN 'debug
```



```

1
2      CONCORDANCE                GENERATE CONCORDANCE
3
4      4/ 14/86
5
100 DEFINT A-Z:ON ERROR GOTO 12000
102 WIDTH "LPT1:",255
105 DIM S$(100)
108 P1$=CHR$(27)+"-"+CHR$(1):P2$=CHR$(27)+"-"+CHR$(0) 'begin and end underscore
110 CLS:PRINT "CONCORDANCE
120 PRINT :INPUT "ENTER FILENAME FOR SELECTED WORD LIST [OMIT .SEL EXTENSION] ",
F$
130 IF F$="" THEN END
135 IF INSTR(S$,".") THEN BEEP:GOTO 120
140 OPEN "I",1,F$+".SEL"
145 PRINT
150 INPUT "ENTER SENTENCES FILENAME [OMIT .SEN EXTENSION] [RETURN TO START] ",S$
155 IF S$="" THEN 190
160 IF INSTR(S$,".") THEN BEEP:GOTO 150
170 OPEN "I",2,S$+".SEN":CLOSE 2:S=S+1:S$(S)=S$:GOTO 150
180 BEEP:PRINT "FILE NOT FOUND!":GOTO 150
190 'PRINT :PRINT :INPUT "ENTER OUTPUT FILENAME [OMIT .CNC EXTENSION] ",O$
195 'IF INSTR(S$,".") THEN BEEP:GOTO 190
197 'OPEN "O",3,O$+".CNC"
200 'BEGIN
205 GOSUB 1000
210 WHILE NOT EOF(1)
220     LINE INPUT #1,W$:W$=LEFT$(W$,25)+" ":W$=LEFT$(W$,INSTR(W$," ")):
UP$=W$:GOSUB 17000:W$=" "+UP$
225     IF LC>52 THEN GOSUB 1000
226     LPRINT "  "P1$;W$;P2$:LPRINT:LC=LC+2:PRINT W$
230     FOR I = 1 TO S
240         OPEN "I",2,S$(I)+".SEN"
245         PRINT "          "S$(I)
250         WHILE NOT EOF(2)
260             LINE INPUT #2,UP$:GOSUB 17000:S$=" "+UP$+" ":L=1
270             P = INSTR(L,S$,W$):IF P=0 THEN 300
280             P$=SPACE$(60)+S$+SPACE$(60):P=P+60
282             IF LC>59 THEN GOSUB 1000
290             LPRINT S$(I)":  " TAB(11) MID$(P$,P-54,54)" "P1$;
MID$(W$,2,LEN(W$)-2);P2$;" ";MID$(P$,P+LEN(W$),55):
LC=LC+1:L=P-59:GOTO 270
300         WEND
310     CLOSE 2
320     NEXT
330     LPRINT :LPRINT:LC=LC+2
340 WEND
350 LPRINT CHR$(12)
990 END
1000 '***** PAGE HEADING *****
1010 PAGE=PAGE+1:IF PAGE>1 THEN LPRINT CHR$(12);
1020 LPRINT CHR$(18)"      CONCORDANCE LIST                "DATE$,TIME$,"PAGE"PAGE
1030 LPRINT CHR$(15)
1040 LPRINT
1050 LC=3
1060 RETURN
12000 '***** ERROR TRAPPING *****
12010 IF ERR=53 AND ERL=170 THEN RESUME 180
12999 ON ERROR GOTO 0:STOP
17000 '***** UPPERFY      UP$ *****

```

```
17010 'INPUT AND OUTPUT:  UP$  
17020 FOR U = 1 TO LEN(UP$):UP=ASC(MID$(UP$,U,1)):IF UP>96 AND UP<123  
      THEN MID$(UP$,U,1)=CHR$(UP-32)  
17030 NEXT:RETURN
```

```

1
2      PRINTER                PRINT FILES ON PRINTER
3
100 DEFINT A-Z:ON ERROR GOTO 12000:WIDTH "LPT1:",255
105 DIM A$(100)
107 CLS:PRINT "PRINTER":PRINT :PRINT :INPUT "COMPRESSED PRINTING? ",A$:IF A$="Y"
    THEN LPRINT CHR$(15);
108 PRINT
110 INPUT "ENTER FILE TO PRINT [INCLUDE FILE EXTENSION] [RETURN TO START] ",A$
115 IF A$="" THEN 200
120 OPEN "I",1,A$:CLOSE 1
130 A=A+1:A$(A)=A$:GOTO 110
200 'PRINT
205 PRINT :PRINT "PRINTING...."
210 FOR I = 1 TO A
215     PRINT A$(I)
220     OPEN "I",1,A$(I):PAGE=0:GOSUB 1000
230     WHILE NOT EOF(1):LINE INPUT #1,A$:LPRINT A$:LC=LC+1:
        IF LC>57 THEN GOSUB 1000
240     WEND
250     CLOSE #1:LPRINT CHR$(12);
260 NEXT
280 END
1000 'NEW PAGE
1005 PAGE=PAGE+1:IF PAGE>1 THEN LPRINT CHR$(12);
1010 LPRINT A$(I),,,DATE$ "TIME$ " PAGE"PAGE
1020 LPRINT
1030 LC=3
1040 RETURN
12000 'ERROR TRAPING
10 IF ERR=53 AND ERL=120 THEN BEEP:PRINT "FILE NOT FOUND.":RESUME 110
    ON ERROR GOTO 0:STOP

```

#:

symbols:

DICTIONARY = list of symbols. Each symbol is a word. Sorted alphabetically

XLAT-TABLE = list of translation entries

a symbol could have the following properties defined:

GRAMCODES = grammar codes

FRAME = frame

XLAT = list of translation entries where the symbol appears in the FROM part

ALIAS = list of translation entries where the symbol appears in the TO part

all symbols have the property LIST-POINTER defined.

the value of this property is the cons-cell in dictionary whose car is the symbol itself

that is (eq symbol (car (get symbol 'list-pointer))) is true for all words.

!#

```
(setq default-gramcodes '(N))
```

```
(setq word-is-in-dictionary nil)
```

```
(setq *unsaved-changes* nil)
```

```
(defun init-dictionary (setq dictionary '(a) xlat-table nil))
```

```
(defun write-dict nil ;write all dictionary data to file
```

```
(let ((dict-s (open "DICTIONARY" :direction :output))
```

```
(xlat-s (open "XLAT-TABLE" :direction :output)))
```

```
(setq *print-length* 200)
```

```
(setq *print-level* 10)
```

```
(mapcar #'(lambda (x) (write-word x dict-s)) dictionary)
```

```
(close dict-s)
```

```
(pprint xlat-table xlat-s)
```

```
(close xlat-s))
```

```
(setq *unsaved-changes* nil))
```

```
(defun write-word (word stream)
```

```
(format stream "~s~%" (cons word (get word 'gramcodes))))
```

```
(defun write-entry (entry stream)
```

```
(format stream "~s~%" entry))
```

```
(defun read-dict nil
```

```
(let ((dict-s (open "DICTIONARY")))
```

```
(setq dictionary
```

```
(do ((l (read dict-s nil nil)
```

```
(read dict-s nil nil))
```

```
(dictionary nil))
```

```
((null l) (reverse dictionary))
```

```
;if done, return rev.
```

```
(setq dictionary (cons (car l) dictionary))
```

```
;prepend word
```

```
(setf (get (car l) 'gramcodes) (cdr l)))
```

```
;gramcodes
```

```
(close dict-s)
```

```
(maplist #'put-pointer dictionary))
```

```
;LIST-POINTER
```

```
(setq *unsaved-changes* nil)
```

```
('((length dictionary) words))
```

```
(defun read-xlat nil
```

```
(let ((stream (open "XLAT-TABLE")))
```

```

    (setq xlat-table (read stream))
    (mapcar #'add-entry xlat-table))
'((length xlat-table) entries in the translation table))

```

```

(defun create-entry (entry)      ;add translation entry if necessary
  (if (member entry (get (caar entry) 'xlat) :test #'equal) ;if exists
      nil
      (progn
        (add-entry entry)
        (setq xlat-table (cons entry xlat-table))))))

```

```

(defun add-entry (entry)          ;update XLAT and ALIAS
  (mapcar #'(lambda (symbol) (add-entry-to-xlat symbol entry))
    (car entry))
  (mapcar #'(lambda (symbol) (add-entry-to-alias symbol entry))
    (cdr entry)))

```

```

(defun add-entry-to-xlat (symbol entry)
  (setf (get symbol 'xlat) (cons entry (get symbol 'xlat)))
  (make-sure symbol))

```

```

(defun add-entry-to-alias (symbol entry)
  (setf (get symbol 'alias) (cons entry (get symbol 'alias)))
  (make-sure symbol))

```

```

(defun make-sure (symbol)      ;make sure that symbol is in the dictionary
  (unless (get symbol 'list-pointer) (add-word symbol)))

```

```

(defun convert-lex-to-dict nil
  ;convert from LEX.L      -- the old format file
  ;      to DICTIONARY -- the new format file
  (let ((stream (open "lex.1")))
    (eval (read stream))
    (mapcar #'convert-word lexicon)
    (setq lexicon nil)
    (makunbound 'lexicon)
    (close stream)
    (maplist #'put-pointer dictionary)
    (write-dict)))

```

```

(defun convert-word (word-entry &aux word)
  (setq word (read-from-string (car word-entry)))
  (add-word word)
  (setf (get word 'gramcodes) (cdr word-entry)))

```

```

(defun add-word (word)
  (unless (get word 'list-pointer)
    (if
      (after word (car (last dictionary)))
      (let ((cons-cell (list word)))
        (rplacd (last dictionary) cons-cell)
        (put-pointer cons-cell))
      (prog (cons-cell)
        (setf cons-cell (member word dictionary :test #'before))
        (rplacb cons-cell
          (cons word
            (cons (car cons-cell)
              (cdr cons-cell)))))))

```

```
(put-pointer (cdr cons-cell))))))
```

```
(defun add-word1 (word) ;assumes ascending order
  (rplacd (last dictionary) (list word))
  (put-pointer (last dictionary)))
```

```
(defun dmu (&optional iword) ;Dictionary Maintenance Utility
  (highlight 'on)
  (format t "~%
  (highlight 'off)
  (prog (command
    (word (if iword iword (car dictionary))) ;by default start with 1st word
    tword
    line)
  loop1
  (small-menu word)
  (setq command (read-char))
  (format t "~C~%" command)
  (case (toupper command)
    ((#\E) ;Edit word
     (format t "Enter word to replace ~s >" iword)
     (setq tword (read-line)) ;retype for now
     (unless
      (null-string tword) ;blank line
      (prog ((codes (get word 'gramcodes)))
        (setq tword (read-from-string tword))
        (unless
          (eq tword word) ;no change
          (if (delete-word word) ;if deletable
              (add-word2 tword codes))))))
    ((#\G) ;edit Grammar codes
     (setf (get word 'gramcodes)
           (input-gramcodes word (if (get word 'gramcodes)
                                     (get word 'gramcodes)
                                     default-gramcodes))))
    ((#\D) ;Delete this word
     (if (delete-word word)
         (format t "~s deleted.~%" word)
         (format t "~s cannot be deleted~%" word)))
    ((#\N) ;Next word
     (if word-is-in-dictionary
         (setq word (cadr (get word 'list-pointer)))
         (setq word (car (member word dictionary :test #'before)))))
    ((#\P) ;Previous word
     (if word-is-in-dictionary
         (setq word (previous-word word))
         (setq word (car (member word
                                (reverse dictionary)
                                :test #'after)))))
    ((#\J) ;Jump to a different word
     (format t "Enter new word>")
     (setq word (read))
     (format t "~%"))
    ((#\C) ;Create new word
```

```

(format t "Enter new word>")
(setq tword
  (read-from-string
    (string-append
      (read-line)
      " nil")))) ;nil if pressed return
(cond ((null tword) nil) ;forget it
      ((consp (get tword 'list-pointer)) ;already a word
        (format t "~s already exists.~%" tword)
        (setq word tword))
      (t ;go ahead, create
        (add-word2 tword default-gramcodes)
        (setq word tword))))

```

```

((#\A) ;Aliases
  (setq word (alias-menu word)))

```

```

((#\X) ;Xlat
  (setq word (xlat-menu word)))

```

```

((#\T) ;Translation entries
  (setq word (translat-menu word)))

```

```

((#\B) ;Browse semantic network
  (browse word))

```

```

((#\M) ;Modify default grammar code for new words
  (setq default-gramcodes
    (input-gramcodes 'the-default default-gramcodes)))

```

```

((#\S) ;Save everything to disk
  (save-to-disk))

```

```

((#\Q) ;Quit
  (if *unsaved-changes*
    (if (y-or-n-p "
Latest changes have not been saved.
Write to disk before quitting? ")
      (save-to-disk)))
  (format t "~%")
  (return '*thanks*))

```

```

((#\!) ;delete frame for this word and don't make faces!

```

```

((#\?) ;help
  (big-menu word))

```

```

(otherwise (beep))

```

```

);case

```

```

(go loop1)))

```

```

(defun delete-word (word) ;returns NIL if not deletable
                          ;or T if deleted

```

```

  (if (deletable word)
    (prog ()
      (rplacd (get (previous-word word) 'list-pointer)
        (cdr (get word 'list-pointer)))
      (setf (get word 'list-pointer) nil)
      (setf (get word 'gramcodes) nil)
      (setq *unsaved-changes* t)
      (return t))
    nil))

```

```
(defun add-word2 (new-word default-codes)
  (setf (get new-word 'gramcodes)
        (input-gramcodes new-word default-codes))
  (add-word new-word)
  (setq *unsaved-changes* t))
```

```
(defun input-gramcodes (word default) ;returns gramcodes
  (let (gramcodes)
    (if (single default) (setq default (car default)))
    (format t "Enter grammar codes for ~s, [return for ~s] >"
            word
            default)
    (setq gramcodes (string-append "(" (read-line) ")")) ;wrap w/parens
    (if (= (length gramcodes) 2)
        (setq gramcodes (format nil "(~s)" default)))
    (setq gramcodes (read-from-string gramcodes))
    (if (listp (car gramcodes))
        (setq gramcodes (car gramcodes))) ;remove if not needed
    gramcodes))
```

```
(defun small-menu (word)
  (format t "~c" (if (get word 'alias) 26 32))
  (highlight 'on)
  (format t "~s" word)
  (highlight 'off)
  (format t "~c" (if (get word 'xlat) 26 32))
  (format t " ~s ~a ~a<E G D N F J C A X T B M S Q ! ?>"
        (get word 'gramcodes)
        (if (deletable word) "*empty*" "")
        (if (setq word-is-in-dictionary
                (and (listp (get word 'list-pointer))
                     (eq (car (get word 'list-pointer))
                          word)))
            ""
            "*NOT IN DICT* ")))
```

```
(defun big-menu (word)
  (format t "~%current word: ")
  (highlight 'on)
  (format t "~s" word)
  (highlight 'off)
  (format t ", grammar codes: ~s, it is ~adeletable.~a"
```

Menu choices:

```
E Edit this word
G edit Grammar codes of this word
D Delete this word
N Next word alphabetically
P Previous word alphabetically
J Jump to a different word
C Create new word +-----+
A Aliases | choices A and X have both |
X Xlat | been replaced by choice T |
T Translation entries +-----+
F Browse semantic network
M Modify default grammar codes for newly created words
S Save everything to disk
Q Quit DMU
! delete the frame associated with this word. I mean it!
? help, (this menu)
```

word

(get word 'gramcodes)



```

    (if (and (listp (get word 'list-pointer))
              (eq (car (get word 'list-pointer))
                  word))
        ""
        "
    *** THIS WORD IS NOT IN THE DICTIONARY *** ")))

```

```

(defun toupper (char)          ;upperfy character
  (if (and (>= char #\a)
           (<= char #\z))
      (- char 32)
      char))

```

```

(defun null-string (string) (= (length string) 0))

```

```

(defun single (l) (and (listp l)
                        (null (cdr l)))) ;is list of length 1

```

```

(defun before (s1 s2) (string< (symbol-name s1) (symbol-name s2)))

```

```

(defun after (s1 s2) (string< (symbol-name s2) (symbol-name s1)))

```

```

(defun previous-word (word)
  (car (member word (cons nil dictionary) :test #'eq-->)))

```

```

(defun eq--> (word element)
  (eq word (cadr (get element 'list-pointer))))

```

```

(defun next-word (word) (cadr (get word 'list-pointer)))

```

```

(defun put-pointers nil (maplist #'put-pointer dictionary) nil)

```

```

(defun put-pointer (cons-cell)
  (setf (get (car cons-cell) 'list-pointer) cons-cell))

```

```

(defun deletable (word)
  (and (listp (get word 'list-pointer))
        (eq (car (get word 'list-pointer))
            word)                                ;word is in dict.
        (not
         (or (get word 'xlat)                    ;word is empty
             (get word 'alias)
             (get word 'frame)))))

```

```

(defun save-to-disk nil
  (write-dict)
  ;***** WRITE FRAMES TO DISK HERE!!! *****
)

```

```

(defun read-from-disk nil
  (values
   (read-dict)
   (read-xlat)
   ;***** READ FRAMES *****
  ))

```

```

(defun alias-menu (word)          ;load "alias.lsp" to use the real thing
  (format t "Please use choice T for translation entries ~%"

```

word)

```
(defun xlat-menu (word)          ;load "xlat.lsp" to use the real thing
  (format t "Please use choice T for translation entries.~%")
  word)
```

```
(defun highlight (switch)
  (case switch ('on (send *terminal-io* :Set-attribute 15))
    ('off (send *terminal-io* :set-attribute 7))))
```

```
;translat.lsp string translation stuff
```

```
(defun translat-menu (word) ;return current word
  (prog ()
    loop1
    (show-translations word)
    (small-menu-t word)
    (setq command (read-char))
    (format t "~C~%" command)
    (case (toupper command)
      ((#\C) ;Create a translation entry
        (create-translation-entry))

      ((#\D) ;Delete one Xlat entry
        (delete-translation-entry word
          (append (get word 'xlat) (get word 'alias))))

      ((#\N) ;Next word
        (if word-is-in-dictionary
          (setq word (cadr (get word 'list-pointer)))
          (setq word (car (member word dictionary :test #'before))))))

      ((#\P) ;Previous word
        (if word-is-in-dictionary
          (setq word (previous-word word))
          (setq word (car (member word
                                (reverse dictionary)
                                :test #'after))))))

      ((#\J) ;Jump to a different word
        (format t "Enter new word>")
        (setq word (read))
        (format t "~%"))

      ((#\B) ;Browse semantic network
        (browse word))

      ((#\M) ;Modify default grammar code for new words
        (setq default-gramcodes
          (input-gramcodes 'the-default default-gramcodes)))

      ((#\S) ;Save everything to disk
        (write-dict)
        ;***** WRITE FRAMES TO DISK HERE!!! (1 of 2) *****
    )

      ((#\Q) ;Quit
        (return word))

      ((#\?) ;help
        (big-menu-t word))

      (otherwise (beep)))
    (go loop1)))
```

```
(defun small-menu-t (word)
  (setq word-is-in-dictionary
    (and (listp (get word 'list-pointer))
      (eq (car (get word 'list-pointer))
        word)))
  (format t "Translation menu: ")
  (highlight 'on)
  (format t "~s" word)
  (highlight 'off))
```

```

(defun big-menu-t (word)
  (format t "
    Translation menu choices:
    C Add a translation entry
    D Delete a translation entry
    N Next word alphabetically
    P Previous word alphabetically
    J Jump to a different word
    B Browse semantic network
    S Save everything to disk
    Q Quit Xlat menu, return to main DMU
    ? help, (this menu)    ~%"))

(defun show-translations (word)
  (prog ((xlat (append (get word 'xlat) (get word 'alias))))
    (if xlat
      (progn
        (format t "Translation containing ")
        (highlight 'on)
        (format t "~s" word)
        (highlight 'off)
        (format t ":%~%" word)
        (dotimes (i (length xlat))
          (format t "  ~d~c~a ~c ~a~%"
                    (+ i 1)
                    #\tab
                    (words (car (nth i xlat)))
                    26
                    (words (cdr (nth i xlat))))))))))

(defun create-translation-entry nil
  (prog (lhs rhs)
    (format t "Enter word(s) to translate from>")
    (setq lhs (read-from-string (string-append "(" (read-line) ")")))
    (if (null lhs)
      nil
      (prog ()
        (if (consp (car lhs)) (setq lhs (car lhs)))
        (format t "Enter word(s) to translate to ([RETURN] to hyphenate)>")
        (setq rhs (read-from-string (string-append "(" (read-line) ")")))
        (if (null rhs) (setq rhs (hyphenate lhs)))
        (if (consp (car rhs)) (setq rhs (car rhs)))
        (create-entry (cons lhs rhs))))))

(defun delete-translation-entry (word list)
  (if (null list)
    (progn
      (beep)
      (format t "~%"))
    (prog (response)
      (format t "Enter entry number to delete>")
      (setq response (read-from-string (string-append (read-line) " ())))))
      (if (numberp response)
        (if (and (>= response 1)
                  (<= response (length list)))
          (delete-entry1 (nth (1- response) list))
          (beep))
        (if response (beep))))))

(defun delete-entry1 (entry)
  (setq xlat-table (delete entry xlat-table))
  (mapcar #'(lambda (word) (delete-entry2 word entry)) (car entry))
  (mapcar #'(lambda (word) (delete-entry3 word entry)) (cdr entry))

```

```
(setf (get word 'xlat) (remove entry (get word 'xlat))))
```

```
(defun delete-entry3 (word entry) ;remove entry from alias list  
  (setf (get word 'alias) (remove entry (get word 'alias))))
```

```
(defun words (l)  
  (string-left-trim  
    "("  
    (string-right-trim  
      ")"  
      (format nil "~s" l))))
```

```
(defun hyphenate (l) ;returns a list of one element --  
                    ;the atoms of l hyphenated
```

```
(let ((string (format nil "~s" l)))  
  (dotimes (i (length string))  
    (if (eq (aref string i) #\space)  
        (setf (aref string i) #\-) )  
    (read-from-string string)))
```

;still to add:

;

- A character level lexical analyzer

(setq \*ultra-verbose\* nil)

(defun preproc (l)

(if \*ultra-verbose\* (format t "~%"))

(rplacd (last l) '(\*end-of-sentence\*))

(maplist #'handle-cons-cell l)

 (rplacd (nthcdr (- (length l) 2) l) nil) ;get rid of token  
 l)

(defun handle-cons-cell (cons-cell)

(unless (numberp (car cons-cell))

(do\* ((candidates

(remove-if

#'(lambda (entry) (neq (caar entry) (car cons-cell)))

(get (car cons-cell) 'xlat))

(cdr candidates))

)

((endp candidates) nil)

(setq lhs (caar candidates))

(if (they-match (caar candidates) cons-cell)

(progn

(if \*ultra-verbose\*

(perform-an-extremely-verbose-function (car candidates)))

(apply-translation (cdr candidates)

cons-cell

(length (caar candidates)))

(setq candidates nil))))))

(defun they-match (lhs list)

(cond ((null lhs) t) ;end of lhs

((null list) nil) ;premature end of list

((eq (car list)

(car lhs))

(they-match (cdr lhs)

(cdr list)))) ;recurse

(defun apply-translation (rhs list still-left)

;replace STILL-LEFT atoms from LIST by elements of rhs

(cond ((null rhs)

(cond ((&gt; still-left 0) ;delete from old list

(rplacd list (cdr list));delete car

(apply-translation nil list (1- still-left))))))

((&gt; still-left 0) ;substitute, do not insert

(rplaca list (car rhs))

(apply-translation (cdr rhs) (cdr list) (1- still-left)))

((= still-left 0)

(rplacd list (cons (car rhs) (cons (car list) (cdr list)))) ;insert

(apply-translation (cdr rhs) (cdr list) 0))))

(defun perform-an-extremely-verbose-function (entry) ;show each substitution

(format t "~a ~c ~a~%"

(words (car entry))

26

(words (cdr entry))))

```
(defun lexan (string)                                ;assumes no punctuation (esp. , and #)
  (mapcan
    #'make-parse-node
    (preproc
      (read-from-string
        (string-append "(" string ")")))))

(defun make-parse-node (word)
  (prog ((gramcodes (get word 'gramcodes)))
    (if (null gramcodes)
      (progn
        (format t "I don't know the grammar codes for ~s~%" word)
        (dmu word)
        (setq gramcodes (get word 'gramcodes))))
    (return '(((
      ,gramcodes
      (,(format nil "~s" word))          ;"WORD" for now
      (,(if (equal (get word 'gramcodes) '(prep))
        (format nil "~s" word)))
      ,0)))))
```

## Section II Input to Utilities

Our initial goal has been to gain insight into the vocabulary in the tactical air arena, and to determine what type of questions, in what form, a user of the expert system might ask. Using the WORDS utilities, we have generated two lexicons and a concordance based on the following input:

- (1) Sample questions from AFIRMS.
- (2) Sample questions from TASKING.
- (3) Sample questions from dialogues between George Devorshak, SYNERGY, and SRA.
- (4) Sample questions generated in an experiment conducted at SRA, in which the questioners believed that they were querying a completed system, whereas in actuality the answers were generated by a human expert.
- (5) A chapter discussing the Tactical Air Command.

Inputs (1) through (4) are included here.



Copies of inputs (1) - (4) described  
in Section II

### \*\*\* Questions from USAFE TASKING OVERVIEW

#### \*\* HQ USAFE level peace

Is EUCOM tasking appropriate for USAFE?

How should USAFE support EUCOM?

Which units will be tasked?

Is plan or tasking sufficient?

Is the likely outcome worth planned or tasked resources?

Is a logistics concept developed or can one be developed?

How can USAFE respond without augmentation?

Can the selected units be supported?

Is there sufficient augmentation?

Can augmentation be improved?

Can USAFE respond by augmenting forces and support?

Can Logistics support Operations Concept?

Which resources are needed to support the Logistics Concept?

Can the force be sustained?

Can airlift support Logistics requirements?

What are current USAFE movement requirements?

What support can MAC and Military Sealift Command provide from CONUS and theater?

Can ground transport requirements be filled?

What capability can be provided to resupply?

What closure times can be supported?

#### \*\*\* Wing level peace

What resources does the tasking require?

What is the expected annual and quarterly schedule for sortie production?

Can the tasked units respond?

Which aircrews are to be assigned to the flying schedule?

What resources are required/available to support the schedule?

Which aircrews are most in need of training during the coming quarter?

Which aircrews are qualified for alert duty?

Are aircrews adequately briefed and aware of mission?

Are aircraft properly prepared?

What is the daily flying schedule?

What is the generation flow plan?

What are the schedules for AGS personnel and weapons assemblers?

What is the best allocation of maintenance personnel for the coming week?

What is the effect of tasking on the expected annual and quarterly

contract for sortie production?

How many aircraft can be generated for the next schedule?

Is the scheduled aircraft ready for launch?

Can returned aircraft be turned for next take off?

What are the training objectives?

What is the training schedule for quarter/month/week?

Do individuals meet performance standards?

What is current and projected workload?

What will degrade repair performance?

Is the service process satisfactory?

Are munitions requirements satisfied?

Are fuel requirements satisfied?

Are critical supplies available to support tasking?

What is the availability of critical base support services?

Which critical facilities require maintenance?

Which facilities critical to tasking be maintained?

#### \*\*\* HQ USAFE Crises

Which resources are needed at MOB, COB, or FOL to perform mission or tasking?

What corrective actions are necessary?

What is the unit response to tasking?  
Which resources are needed to support sorties required?  
Where should resources be placed?  
Which resources should be augmented?  
Which shortfalls need attention?  
What resource requirement changes and shortfalls affect current plans and tasking?  
Which resources can be resupplied to meet tasking?  
Can the resources be moved?  
What support is needed to ensure missions or taskings are performed?  
What shortfalls are critical?  
What must be moved?  
What alternate transportation can be used?  
What alternate routes can be used?

\*\*\* WING level, crises

Is the ATO feasible?  
What is the daily flying schedule?  
Which aircrews are to be assigned to flying schedule?  
Which weapons are required?  
What are possible limitations to accomplishment of tasking?  
Which aircrews are qualified for alert duty?  
Are aircrews adequately briefed and aware of mission?  
Are aircraft properly prepared?  
Have the missions been properly planned?  
Are aircraft properly serviced?  
Can recovered aircraft be turned to meet next take off?  
Can the tasked units respond?  
What resources are required?  
What is the daily flying schedule?  
What is the generation flow plan?  
What are the schedules for AGS personnel and weapons assemblers?  
How many aircraft can be generated from the next schedule?  
Is the scheduled aircraft ready for launch?  
Can returned aircraft be turned for next take off?  
What are the training objectives?  
What is the training schedule for quarter/month/week?  
Do individuals meet performance standards?  
What is the current and projected workload?  
What will degrade repair performance?  
Is the repair process satisfactory?

\*\* From body of introduction

Why? How did you get that answer?  
How do you know that airplanes fly?

Of all those in that Air Base, how many can fly air to ground missions?  
Of the ones we have in the central region, how many are at Bitburg?

\*\*\* From "Typical XOODIM Question"

What kind of airlift do I have?  
What kind of TAC air am I going to be looking for?  
Where am I going to be looking for it?  
How fast can I get my logistics tail up and running?

\*\*\* From sample questions -- tabular form

Do I have required SLC?  
If we don't have enough built, can we build them?  
How many sorties do I have? SCL's?

Is there anything in the task that will affect sortie generation capability?  
Can they reach target?  
Where is target?  
Is there any extraordinary routing that will affect this?  
Can the aircraft reach home after this mission?

How many mission qualified aircrews do you have?  
How many defense suppression qualified do you have? — ellipsis

Are there any airfield constraints on sortie generation?  
Do I have sufficient runway to launch and recover my aircraft?  
Do I have access via runways?  
What is the status of the arresting gear?

Can you fuel my airplanes? Can you refuel the airplane on time?  
How many can I fuel sheltered?  
If all can't be sheltered, schedule refueling without movement.

Are there any maintenance constraints on sortie production Capability?  
Do I have enough people to launch and recover?

Are there any intelligence factors that would influence sortie production?

How many sorties are needed to man 3 suppression caps from first light to dusk with SCL 1 and 2? X

Whats the transit time to and from each cap point? \*\* note "Whats"  
Whats the flying window?

How long can aircraft remain in each cap?  
What launch and recovery pattern is needed to do the mission?  
What is SCL 1 and 2?

What are the limiting factors within our control which we can change to do the mission?

What are the limiting factors within higher headquarters control that could be changed to do the mission?

How many sorties are needed to man 3 lane cap portions with a minimum of two aircraft from light to dusk with SCL?

Whats the transit time to and from each cap point?  
Whats the flying window?

How long can aircraft remain in each cap?  
What launch and recovery pattern is needed to do this mission?

Are there any constraints on sortie generations?  
Do I have sufficient runway to launch and recover my aircraft?  
Do I have access via taxiways?  
What is status of the arresting gear?

Is there anything in the task that will affect sortie generation capability?  
Can they reach target?  
Where is target?  
Is there any extraordinary routing that will affect this?  
Can the aircraft reach home after this mission?

Do I have required SLC?  
If we don't have enough built, can we build them?  
How many sorties do I have? SCL's?

Do I have enough dumb bombs to support 140 sorties tomorrow?  
If we don't have enough built, can we build them?  
How many sorties do I have? SCL's? \*\*\* note -- have sorties

How many air-to-ground sorties can you produce tomorrow with dumb bombs  
with morning, late afternoon and evening drops without air to air  
refueling?

How many FMC aircraft do we have?

How many FMC aircraft do we have?

How many wargoers do we have? *many* *u*

How many air-to-ground qualified aircrews do you have?

How many qualified people of you have that can do this? *many*

Are there any airfield constraints on sortie generation?

Can we man 3 lane CAP positions with a minimum of two aircraft from  
light to dusk with SCLB?

Can we man 3 suppression caps from first light to dusk with SCL 1 and 2?

\*\*\*\*\* SAMPLE QUESTIONS FROM AFIRMS \*\*\*\*\*

\*\*\* HQ USAFE

What are the trends for the tasked missions relative to one another and over time for USAFE Central Region Forces?

Approximately how many sorties are tasked for Counter Air, CAS/BAI, and Interdiction on Day 5? Day 10?

How many total sorties are required on Day 5? Day 20? Day 30?

What missions are the various Central Region fighters tasked for and how many sorties for each mission?

How many sorties of BAI and CAS are tasked for Central Region A-7D and A10A aircraft?

How many sorties were tasked for all missions and Mission Design Series (MDS)?

What are the trends of MDS sorties in this OPORD relative to one another and time?

Approximately how many sorties are F-4 and F-16 aircraft tasked to fly at Day 1? Day 10?

How many total sorties are required on Day 5? Day 15? Day 30?

What is the integrated resource capability over time to support the tasking with the current assets?

On what day does the capability shortfall first occur?

Approximately how many sorties can be flown on Day 30? What percentage is that of the tasking on Day 10?

What is the capability of aircraft, spares, JP-4, and munitions to support the tasking with current assets?

Which resource constrains first? Second?

If sorties are not flown because of spares does fuel remain as the second constraining factor?

When does the F-4E/G spares shortfall start to occur?

Does the shortfall improve before Day 30?

What are the critical spares for the F-4E/G at Day 15?

Can the F-4E/G drop bombs with inoperative missile-firing circuits and autopilots?

How many tires are needed to satisfy demand?

How many days of AGM-65 are on-hand?

How many of MK-20?

How long will it take to deplete the inventory of AIM-7 missiles on the Central Region?

On what day will the number be depleted?

What is the status of MK-82 at Spangdahlem?

Are there any munitions that are below the threshold level at Spangdahlem?

How many AGM-45 missiles does Spang have?

What are the fuel capability trends for the Central Region air bases with current assets?

When do the shortfalls start to occur?

Which bases are able to support their tasking for the first 30 days?

What is the status of all bases in the Central Region?

What is the runway status at Ahlhorn?

What problem areas exist for Central Region bases?

What status factors are NMC for Central Region bases?

How many F-15 aircraft are at Bitburg and, of those, how many are FMC?

What Central Region bases are having communication problems?

Is there a secure voice line capability at Bitburg?

Are any NMC Radar Approach Controls in the Central Region?

Is there a RAPCON/Ground Controlled Approach that is NMC?

How many fighter/recce aircraft are needed to replace losses?

For each MDS, how many aircraft remain?

How many aircrew members have been lost?

\*\*\*HQ USAF

What missions are the various Central Region fighters tasked for and how many sorties for each mission?

How many sorties of BAI and CAS are tasked for Central Region F-16A and A10-A aircraft?

How many sorties were tasked for all missions and Mission Design Series?

What are the trends of MDS sorties in this OPORD relative to one another and over time?

Approximately how many sorties are F/RF-4 and F-16 aircraft tasked to fly at Day 1? Day 10?

How many total sorties are required on Day 5? Day 15? Day 30?

Which MDS is the first to be constrained by spare parts?

Does the shortfall improve before Day 30? By Day 120?

Will theatre jet fuel sustain operations until CONUS resupply arrives?

Will the CONUS resupply sustain operations?

What are the fuel sustainability trends for the Central Region air bases with current assets?

When do shortfalls start to occur?

Which bases are able to support their tasking for the first 30 days?

How soon can the munitions from outside sources support USAFE?

How long before the munitions can sustain the task?

How is the sortie requirement impacted with substitute munitions to maintain the target kill effectiveness of the preferred munition?

Can munitions support the sortie requirement?

What is the current allocation of air-to-air missiles?

What is the effect on the commands losing missiles in the proposed reallocation?

Which Central Region bases have problems?

How many aircraft does Spangdahlem have?

Which Central Region bases have fuels problems?

What is causing Spang's fuel problem?

Which bases have problems and what are their problem areas?



What status factors are NMC for Central Region bases?

How many F-15 aircraft are at Bitburg and, of those, how many are FMC?

How are the current fighter/recce attrition trends changing with respect to the overall trends?

How many fighter/recce aircraft have been lost to date?

How many fighter/recce pilots were lost yesterday?

How many fighter/recce aircraft have been lost?

For each MDS, how many aircraft remain?

How many aircrew members have been lost?

What is the DG objective for each type of munition?

What submunitions are included in each category?

In what year will the Air Force be able to meet its munitions procurement objectives in each of the mission areas?

\*\*\* Wing

What are the details about mission number 1fS241?

What missions are tasked with Standard Conventional Load UD0620?

How are the tasked sorties divided by mission type for the Wing?

How many BAI sorties were tasked?

What is the required turn pattern to meet the ATD?

What is the minimum number of aircraft required for the tasking?

Can the Wing fill the tasked munitions requirements specified in the ATD?

If there is a shortfall, which munition(s) is the problem?

How many MK-82s were specified in the tasking?

Are there any munitions shortfalls? When?

Are there any rack configuration problems?

How many AGM-65s will be needed by launch?

What targets are behind the Forward Edge of Battle Area?

Is the approximate distance to the target within range of the aircraft fuel tank configuration?

Are any targets in areas requiring Border Crossing Authority?

What is the weather at the alternates?

What is the capability of the Wing relative to tasking?

\* i.e. for each (potentially limiting) factor, how many sorties can I generate,  
\* given that the tasking said to generate 110 sorties.

If there is a shortfall, in what area does it occur?

Can the Wing aircraft support the task?

How much is the F-4G shortfall and in which launches does it occur?

How many more sorties can the F-4E support?

How many mission capable F-4E aircraft will be available at 0600Z, 20 May 83?

Of the Non-Mission Capable aircraft, which category is the largest?

What is the reason for so many NMC F-4G aircraft?

\* e.g. supply, maintenance, or both?

What is the supply requisition status of parts for each NMCS or Non-Mission Capable Both aircraft?

Which squadron has been most affected by NMCS or NMCB?

How many 370-gallon and 600-gallon fuel tanks are built up?

How many days supply of fuel tanks remain?

How many days supply of TRAP remain if they are consumed at a given rate per day?

What is the amount of TRAP on hand?

TRAP = Tanks, Racks, Adaptors, and Pylons

How much of the ATO can the aircrews support?

How many more air defense missions could the aircrews support?

What MR aircrews are available by mission type?

What MR aircrews are available by squadron?

Can the ATO be supported by the Fuels Branch?

At the present level of tasking, how many days of JP-4 are on-hand?

How much excess capability is there in the refueling area?

How much of the ATO can Refueling support?

\* note that Refueling is somewhat different from refueling

How much excess capability is there?

Are the Hot Pits required to support the ATO?

What is the availability of R-14 refueling trucks?

If needed, how many Hot Pits are available?

What are the peak hours of operation for the refueling trucks?

Are there enough drivers to man the refueling trucks for 24 hour operation?

How much of the ATO can Munitions support?

\* Here again, Munitions has a different connotation from munitions

\* ie the department vs the commodity, but this if of little or no

\* consequence.

How much is the munitions shortfall and what is causing it?

How many of the tasked sorties munitions are already built?

Of those already built, how many are in shelters?

How much of the ATO can the munitions load crews and equipment support?

If there is a munitions loading shortfall, what is causing it?

How much of the ATO can munitions distribution support?

If there is a munitions distribution shortfall, what is causing it?

How much of the ATO can the munitions assembly process support?

If there is a munitions assembly shortfall, what is causing it?

Are there any alternate munitions available to use in lieu of the short munition?

Are there any AIM-9J missiles stored off-base?

Are there any equipment shortages?

Are there enough 5/10-ton tractors to substitute for the shortage of MB-4 Bobtail trucks?

How many forklifts are awaiting maintenance and parts?

How many load crews are needed for the first shift?

How many load crews are available as of 0600Z 20 May 83?

What are the SCL qualifications load crews #1?

Are there any extra load crew members?

Can the specific tasking level be supported by maintenance?

If there is a maintenance support shortfall, what is causing it?

How much of the ATO can the AGE support handle?

If there is an AGE support shortfall, what is causing it?

What is the status of the Wing AGE availability?

How many MD-3's are Operationally Ready?

What is the schedule mission flow?

What is the required turn pattern to meet the modified tasking?

What is the minimum aircraft required for the modified tasking?

What is the approved flight schedule?

What mission is on line #18 of the flying schedule?

How many aircraft will be Mission Capable in time to meet tasking?

What are the ETIC and problems with NMC aircraft?

Where are the aircraft located?

Once the load out begins, what lines are not meeting the schedule?

If a line is not on schedule, which part of the line's schedule is having difficulty?

What is the overall condition of the aircraft flying process?

What is the condition code of the aircraft?

Where is the aircraft?

What is the status of the aircraft?

What type of munitions is the aircraft configured with?

When did the aircraft land?

When is the aircraft scheduled to take off?

How close is the aircraft to takeoff?

What is the Estimated Time in Commission for code 3 aircraft?

What is the status of major components in the POL system?

How many fillstands are available?

How many fuel trucks are available? Fuel truck drivers?

How many hydrants are available?

\*\*\*\*\*  
05-15-1986 12:43:05  
Q: Hi hatte. Please say something  
A: can you hear me  
Q: Yes. say something else  
A: hello now is the time for George to walk in  
Q: how many sorties?  
\*\*\*\*\*  
05-15-1986 12:54:08  
Q: Can I fly BAI with my chickens?  
A: yes if you fly wargoose  
Q: what are my oca capabilities  
A: i do not understand the question; please try again.  
Q: how many oca sorties can we generate in the next three days  
A: 1324 sorties.  
Q: what is the daily oca sortie generation rate for the next three days  
\*\*\*\*\*  
05-15-1986 13:12:28  
Q: what is my daily oca sortie generation rate  
A: 1.75  
Q: how many sorties per day  
A: 476  
Q: which aircraft  
A: they are f-4e and f-16.  
Q: how many f-4e oca sorties in the next three days  
A: 441  
Q: how many f-4e oca sorties per day  
A: i don't understand the question; please try again.  
Q: how many oca sorties per day  
A: 476  
Q: how many f-4e oca sorties per day  
A: i don't understand the question; please try again.  
Q: how many f-4e sorties can i generate in the next three days  
A: 441  
Q: how many f-4e oca sorties can i generate in the next three days  
A: 441  
Q: are these all oca  
A: yes.  
Q: what is the limfac  
A: i don't understand the question; please try again.  
Q: why cant i generate more sorties  
A: there are no more f-4e aircraft.  
Q: how long can i maintain a 1.75 sgr  
A: what is the meaning of "sgr"?  
Q: sortie generation rate  
A: 5 days.  
Q: why cant i go past five days  
A: the limiting factor is oca munitions.  
Q: which munitions  
A: general purpose bombs.  
Q: what is the limiting factor  
A: the limiting factor is oca munitions.  
Q: how many days fuel do i have  
A: 6 days.  
Q: clarify 6 days

A: i don't understand the question; please try again.

Q: how many days fuel do i have for f-4e oca sorties  
A: 6 days.

Q: How many f4s foc at hahn?

A: i don't understand the question; please try again.

Q: How many f4s pmc at hahn?

A: 0

Q: how about fmc?

A: 0

Q: are there any f4s at hahn

A: no.

Q: how many f15 at hahn

A: 0

Q: how about f16s?

A: 73

Q: how many fmc

A: 55

Q: and pmc?

A: 11

Q: what is the status of the other7?

A: i don't understand the question; please try again.

Q: what is the status of the other 7 f16s?

A: nmc.

Q: What is the sum of fmc, pmc, and nmc?

A: 73

Q: How much pol at spang ?

A: 1700000 gallons.

Q: what kind?

A: JP-8.

Q: how many pounds would that be?

A: i don't understand the question; please try again.

Q: how many pounds to a gallon of jp8?

A: 6.4 pounds per gallon

Q: what is surge sortie rate for f4s at spang?

A: 1.75

Q: so how many sorties can hahn generate per day?

A: 126

Q: how many f4s at spangdahlem?

A: 72

Q: how did you get so damn smart?

A: i don't understand the question; please try again.

Q: how many cas msns can hahn generate in first 5 days?

A: what is the meaning of "msns"?

Q: missions

A: hahn does not support cas missions.

Q: how about wild weasel sorties

A: hahn does not support wild weasel missions.

Q: how about spang

A: 674

Q: how many the first day

A: 135

Q: how many the 2nd day

A: 135

Q: give me the number of sorties that hahn can generate for each of the first 5 days

A: (1 128) (2 128) (3 128) (4 117) (5 117)

Q: thank you

A: thank you.

Q: Michael here,; say something

A: yo

Q: Hey Jim, What the hell is my username on the Symbolics? (It is booted)

A: you don't have one yet.

Q: So What's yours?

A: this is bob; jim says the machine is in deep kimshi.

\*\*\*\*\*

05-15-1986 14:19:28

Q: what can you tell me.

A: tactical air capabilities in central europe.

Q: how many or airplanes do I have?

A: what is the meaning of "or"?

Q: Operationally ready

A: 221

Q: munitions status

A: please ask a question.

Q: What kind of airplanes do I have?

A: f-4e, f-4g, f-15, f-16.

Q: What type of munitions for f-16

A: i don't understand the question; please try again.

Q: Do you have munitions in file?

A: yes: smart-munitions and dumb-munitions.

Q: list the smart-munitions

A: agm-65 (maverick).

Q: list the dumb-munitions

A: mk-82 (general-purpose bomb), 20mm (hei and api).

Q: Do I have special equipment on the f-16 for agm-65?

A: yes: the lau-45 launcher.

Q: How many agm-65 do I have?

A: f-16 is at hahn and ramstein; how many agm-65 at those bases?

Q: How many agm-65's do I have at Hahn and Ramstein?

A: 310

Q: What is my planned or actual aircraft attrition rate?

A: planned attrition rate is 0.05

Q: How many f-16s do I have at Hahn and Ramstein?

A: 146

Q: How many at Hahn.

A: 73

Q: At 3 sorties a day for Hahn and Ramstein how long will it be before my agm-65's go to 0?

A: 103 days.

Q: Given attrition rate how long will it be before I will have 50 airplanes at Hahn and 50 airplanes at Ramstein?

A: plans call for augmentation at 60 aircraft.

Q: Given the aircraft attrition rate how long will it be before I have 50 airplanes at Hahn?

A: plans call for augmentation at 60 aircraft.

Q: When will I reach 61 airplanes at Hahn with the attrition rate of 0.05?

A: on day 4.

Q: What are my airfield conditions?

A: Hahn is operational.

Q: How long is the runway

A: 9800 feet.

Q: What is the POL status at Hahn?

A: 1700000 gallons.

Q: What are the weather conditions?



A: current weather conditions are not known.

Q: What's my aircraft recovery rate?

A: 60 aircraft per hour capability.

Q: Is that visual flight rules (vfr)?

A: current weather conditions are not known.

Q: What is the status of my aircrews?

A: hahn aircrews are operational.

Q: How many do I have?

A: 105

Q: How many pilots do I have?

A: 105

Q: Are they combat ready?

A: yes.

Q: Do any crews need specialized training?

A: i don't know about individual crews.

Q: Are they current with their training requirements?

A: i don't know about individual crews.

Q: What is my pilot attrition factor?

A: 1 per aircraft.

Q: What type of aircrew is in the back seat?

A: f-16 is a single-seat aircraft.

Q: What type of aircrew is in the back seat of an f-4e?

A: weapons-system-operator.

Q: Are they combat ready?

A: i don't know about individual crews.

Q: How many do I have?

A: 64

Q: How many f-4e's do I have?

A: 48

Q: How many pilots?

A: 105

Q: How many f-4's do I have?

A: 72

Q: What is the ratio between f-4 pilots and f-4 aircraft?

A: 1.25

Q: What is the ratio between f-4 weapons-system-operators and f-4 aircraft?

A: 1.25

Q: What type of f-4 airplanes do I have, and how many of each type?

A: (f-4e 48) (f-4g 24)

Q: What is my sortie generation capability?

A: which airbases?

Q: Hahn

A: 135 day 1-3, 117 day 4-

Q: What kinda airplanes at Hahn?

A: i don't understand the question; please try again.

Q: What airplanes at Hahn?

A: f-16.

Q: Where are the f-4's located?

A: spangdahlem.

Q: Sortie generation at Spang

A: i don't understand the question; please try again.

Q: Sortie generation at Spangdahlem

A: i don't understand the question; please try again.

Q: Sortie generation at Spangdahlem?

A: i don't understand the question; please try again.

Q: What is the sortie generation at Spangdahlem

A: 128 day 1-3, 102 day 4-

Q: based upon a sortie rate of 10 per day at Spangdahlem when will I run out of dumb-munitions?

A: day 5.

Q: Can the aircraft perform nuclear operations?

A: i don't know about "nuclear" operations.

Q: Can the aircraft perform conventional missions?

A: yes.

Q: What kind of conventional missions?

(f-4g sead) (f-4e sead, oca, ai, dca)

what is sead?

A: supression of enemy air defense.

Q: What is oca, ai, dca?

A: offensive counter-air, air interdiction, defensive counter-air.

Q: What weapons do I need for ai?

A: agm-65 or mk-82.

Q: for oca?

A: agm-65 or mk-82.

Q: for dca?

A: aim-7, aim-9, 20mm.

Q: How many of these in stock?

A: (aim-7 112) (aim-9 220) (20mm 1000000)

Q: What is the expenditure rate per sortie of 20mm.

A: 300 rounds per sortie.

\*\*\*\*\*

05/16-1986 13:12:45

Q: how many f-4s are flyable at spangdahlem now?

A: i'd say about 58.

Q: who is "i'd"?

A: me. But since it's Friday, we'd only have 52.

Q: how many fmc, pmc, wargoer by risk out of that 58?

A: 39 fmc; 6 pmc; 2 low risk; 2 moderate; and 9 high risk.

Q: good, how many sorties will that get me on a 3 wave day?

A: 102

Q: that's lousy. what can we do to get sortie production up to 142?

A: I didn't use high risk a/c. I could get 15 more sorties if I use high risk.

Q: 15 plus 102 is 117. so answer my question or i will condemn you to eternal damnation.

A: I can't make 142 unless I kill my crews for the next 2 days

Q: i'm satisfied with that. now, do you want to stay with me, or should i send you back to sherman?

A: keep on going, George.

Q: o.k., give me the same flying profile for hahn, bit, and spang.

A: for Hahn, 130 sorties; for Bit, 122 sorties; I already gave you Spang.

Q: right, i meant bitburg.

A: Ramstein, right? I can only give you 32 from Ramstein.

Q: thanks, is that also based on a 3 day go?

A: yes

Q: o.k., with 1 squadron rolled dca at hahn, how many sorties per go can we get from hahn tomorrow for bai?

A: That's tough. I think we could only get about 30 per go.

Q: that's enough. how many sead pairs can we get for each wave to support hahn?

A: we could only get 4 pairs and that's stretching it.

Q: good. now if i lash-up 8 eagles with each package, how many will i have left at bit on 5 minute alert?

A: We'd have about 4.

Q: do you have your oplan

A: yes

Q: how many are called for

A: how amany what?

Q: sorties for tomorrow

A: 520 sorties.

Q: what % air to air, air to ground

A: 60% air-to-air, 40% air-to-mud

Q: which has priority

A: air-to-air. If we don't keep our bases open, we're dead.

Q: do you mean dca or oca

A: dca

Q: is there a significant sam threat

A: yes, and other threats

Q: within range of my dca orbits

A: only the SA-5s.

Q: what is the weather in your operational area

A: bad, cloudy, drizzle, but we can fly.

Q: can you fully support dca

A: no

Q: did you plan to use war-goers

A: no

Q: how many can you generate tomorrow, that is dca sorties

A: 210 sorties tomorrow.

Q: plan to use war-goers with each flight having 1 radar, operational out to 45 nm range and 1 fmc ecm pod / flight

A: ok

Q: how many of those can you generate

A: I can get an additional 30 sorties tomorrow.

Q: how many are you short of whats really needed, your personal assessment

A: I've got 4 compressors that are hard broke.

Q: i mean how many dca sorties am i short

A: about 20.

Q: do you plan to launch waves, if so how many

A: 3 waves per day

Q: do you think you can make up the difference by using continuous launch. Assume maintenance can support this

A: no, I'd kill my crews in two days.

Q: are crews the most constraining factor

A: yes

Q: maint or flt crews

A: flight crews

Q: can any other base or bases cover the 20 needed dca sorties

A: no

Q: will additional ar help, assume longer sorties

A: what's "ar"?

Q: air refueling. also denoted as a/r

A: no, my crews will drop out of the sky; they just can't fly that long.

Q: whats implied is, flying longer sorties implies less crews used

A: gotcha--new answer: "I could get 30 additional sorties if I use a/r"

Q: good, execute that for dca. what resources do you most urgently need in the next three days

A: AIM-9Ls. I'm going to run out of them in two days.

Q: will 600 help, if i get you 200 tomorrow and 400 in 5 days

A: I'm using almost 200 a day. It'll be close.

Q: do you realize that humans get tired. break break

A: thank you, Lou.

### Section III Output from Utilities

Inlcuded in this section are:

- (1) vocab.mrg, an alphabetical listing of the vocabulary generated from the input in Section II.
- (2) vocab.num, a listing by frequency of the same vocabulary.
- (3) a concordance based on some sample questions.
- (4) a concordance based on all the inputs described in Section II.

vocab.mrg (alphabetical)

"	1
"	1
"D"	1
"ILLUMINATION"	1
"LOOK	1
"TILL	1
-	1
000-FEET	1
000-POUND	4
000-POUND-CLASS	1
0600Z	2
10-TON	1
A	1
11A	1
150-	1
1960S	1
1970S	1
1FS241	1
20-	1
20-MILLIMETER	1
230-GALLON	1
24-HOUR	1
24TH	1
2ND	2
30-	1
30-MILLIMETER	1
30-GALLON	1
41ST	1
440-POUND	1
4G	1
500-POUND	6
552D	4
6-INCH	1
600-GALLON	1
62-MILLIMETER	2
6ACCS	1
6TH	1
707-320B	1
7D	1
7TH	3
800-POUND	1
82'S	1
844-POUND	1
87B	1
89B	4
8TH	1
960TH	1
961ST	1
97S	1
9E	1
A	155
A	1
A-10	8
A-7	2
A-7D	6

A	1
A-A	1
A-G	1
A10-A	1
A10A	1
AAA	1
ABCCC	3
ABILITY	6
ABLE	6
ABOUT	2
ABOVE	3
ABSENCE	1
ABSOLUTE	1
ACADEMIC	1
ACADEMY	1
ACC	7
ACCEPTABLE	1
ACCESS	2
ACCOMMODATE	1 -
ACCOMPANY	1
ACCOMPANYING	1
ACCOMPLISH	1 -
ACCOMPLISHED	3
ACCOMPLISHES	1
ACCOMPLISHING	1
ACCOMPLISHMENT	1
ACCORDED	1 -
ACCORDING	2
ACCS	1
ACCS'	1
ACCURACY	4
ACCURATE	3
ACHIEVE	4 -
ACHIEVED	6
ACHIEVING	1
ACQUIRE	1 -
ACQUIRING	2
ACQUISITION	3
ACT	1 -
ACTING	2
ACTION	8
ACTIONS	2
ACTIVE	9
ACTIVITIES	5
ACTIVITY	2
ACTUATE	1
ADAPTATION	1
ADAPTED	1
ADDED	1 -
ADDITION	6
ADDITIONAL	4
ADJUVANT	2
ADJACENT	1
ADJUST	1 -
ADJUSTMENT	1

ADOC	1
ADVANCE	2 -
ADVANCED	5
ADVANTAGE	1
ADVERSE	4
ADVERSE-WEATHER	1
ADVISE	1 -
ADVISING	1
AERIAL	8
AEROMEDICAL	2
AEROSPACE	8
AFCC	1
AFFECT	7 -
AFFECTED	1
AFFECTING	1
AFFORDED	1
AFLC	1
AFSC	1
AFT	2
AFTEC	2
AFTER	4
AFTERNOON	3
AGAINST	20
AGE	3
AGENCIES	1
AGENCY	3
AGGRESSION	2
AGGRESSIVE	1
AGGRESSOR	3
AGM-45	8
AGM-65	2
AGM-65A	1
AGM-65D	1
AGM-65S	1
AGM-78	5
AGM-88	3
AGS	2
AHLHORN	1
AI	1
AID	1
AIDS	1
AIM	1
AIM-4F	4
AIM-4G	1
AIM-7	3
AIM-7E	2
AIM-7F	2
AIM-9	3
AIM-9J	1
AIM-9L	1
AIM-9P	1
AIR	279
AIR-	2
AIR-2A	4
AIR-BREATHING	1



AIR-TO-LAND	2
AIR-TO-	1
AIR-TO-AIR	11
AIR-TO-GROUND	11
AIRBORNE	21
AIRCRAFT	126
AIRCRAFT'S	2
AIRCRAFT-REMOTELY	1
AIRCREW	8
AIRCROWS	19
AIRDROP	2
AIRFIELD	4
AIRFRAME	1
AIRLAND	2
AIRLANDING	1
AIRLIFT	16
AIRPLANE	1
AIRPLANES	2
AIRPOWER	2
AIRSPACE	5
AIRSTRIKE	1
ALERT	3
ALERTING	2
ALFA	1
ALL	30
ALL-WEATHER	4
ALLOCATED	2
ALLOCATION	2
ALLOW	1
ALLOWED	1
ALLOWS	3
ALOC	1
ALONE	1
ALREADY	2
ALSO	18
ALTERNATE	4
ALTERNATES	1
ALTOUGH	3
ALTITUDE	4
ALTITUDES	4
AM	2
AMERICA	2
AMERICAN	3
AMOUNT	1
AMPHIBIOUS	2
AMRAAM	1
AN	54
ANALYSIS	1
AND	539
ANG	6
ANNUAL	1
ANNUAL	2
ANOTHER	5
ANS	1

ARMOR	1
ARMOR	2
ANTIPERSONNEL	1
ANTIRADIATION	3
ANTISUBMARINE	1
ANTITANK	1
ANY	25
ANYTHING	3
ANYWHERE	1
AP	1
APPARENT	1
APPLICABLE	1
APPLICATIONS	1
APPLIED	1 /
APPORTIONMENT	1
APPROACH	3 /
APPROPRIATE	6
APPROVED	1
APPROXIMATE	1
APPROXIMATELY	7
APR-38	1
ARE	194 /
AREA	22
AREAS	13
ARES	1
ARIZONA	1
ARM	3
ARMAMENT	5
ARMED	2
ARMOR	1
ARMORED	1
ARMS	1
ARMY	8
ARN-101	3
ARRANGEMENTS	1
ARRESTING	2
ARRIVAL	1
ARRIVES	1
ARSENAL	1
AS	54
ASIA	2
ASOC	3
ASPECTS	1
ASRT	1
ASSAULTS	1
ASSEMBLERS	2
ASSEMBLY	2
ASSESS	1
ASSESSMENT	1
ASSET	1
ASSETS	9
ASSIGNED	17
ASSIGNMENT	2
ASSIST	2 /
ASSISTANCE	1

ACTIVISTS	1
ADJACENT	2
ASSUME	1 -
ASSURE	1
ASSURED	1 -
ASSURES	1
AT	58
AT-38	3
AT-38B	2
ATC	1
ATD	1
ATLANTIC	2
ATO	12
ATTACH	7 -
ATTACHED	1
ATTACHMENT	1
ATTACK	14 -
ATTACKED	1
ATTACKS	1
ATTAINMENT	2
ATTEMPT	1
ATTEMPTING	1
ATTENTION	1
ATTRITION	2
AUGMENT	2 -
AUGMENTATION	5
AUGMENTED	1
AUGMENTING	1
AUGMENTS	1
AUTHORITATIVE	1
AUTHORITIES	1
AUTHORITY	6
AUTHORIZE	1 -
AUTHORIZED	1
AUTOPILOTS	1
AUXILIARY	1
AVAILABILITY	4
AVAILABLE	19
AVIONICS	3
AVOID	2 -
AWACS	8
AWAITING	1
AWARE	2
B	1
BACK	1
BAI	8
BASE	19
BASEBALL-SIZE	1
BASED	5 - ?
BASES	15
BASIC	5
BATTLES	3
BATTLE	14
BATTLEFIELD	6
BAY	3

BY	1
BE	100
BEACON	1
BEAR	1
BECAUSE	1
BEEN	11
BEFORE	6
BEGAN	1
BEGINS	1
BEHAVIOR	1
BEHIND	2
BEING	8
BELOW	5
BENEFICIAL	1
BERGSTROM	1
BEST	1
BETTER	1
BETWEEN	2
BEYOND	1
BIRD	1
BITBURG	5
BLU-97	1
BLUE	1
BLUNTS	1
BOBTAIL	1
BOMB	9
BOMBER	2
BOMBERS	2
BOMBLETS	5
BOMBS	13
BOOK	1
BOOSTING	1
BORDER	1
BOTH	15
BOUNDARIES	2
BOWING	1
BRANCH	1
BRAND	1
BREAKAGE	1
BREAKTHROUGH	1
BRIEFED	2
BRIEFLY	1
BROAD	2
BROUGHT	1
BRUS	1
BU	1
BUILD	3
BUILDING	1
BUILT	7
BUT	6
BY	70
C-130	1
C-135	1
C2	2

	3
C3I	1
CALL	3 -
CAMERA	1
CAMERAS	1
CAMOUFLAGE	1
CAMPAIGN	8
CAN	93 -
CAN'T	1
CANISTER	2
CANNON	6
CANNONS	1
CANNOT	1
CAP	7
CAPABILITIES	12
CAPABILITY	46
CAPABLE	20
CAPACITY	2
CAPS	2
CAPSULE	1
CAREFUL	2
CAREFULLY	1
CARGO	1
CAROLINA	1
CARRIED	10
CARRIER	1
CARRIES	3
CARRY	8
CARRYING	2
CAS	7
CATEGORIES	3
CATEGORY	3
CAUSE	1
CAUSING	7
CBU	1
CBU-52	2
CBU-58	2
CBU-87	1
CBU-87B	1
CBU-89	4
CEILING	4
CEM	2
CENTER	25
CENTERLINE	2
CENTERS	1
CENTRAL	19
CENTRALIZED	9
CENTRALLY	1
CERTAIN	1
CHAFF	2
CHALLENGE	1
CHANGE	1
CHANGED	1
CHANGES	3
CHANGING	2

QUARTER	6
QUARTERS	1
CHARACTERISTIC	2
CHARACTERISTICS	3
CHARGE	2
CHARGED	2
CHIEF	8
CHOICE	1
CHOOSE	1
CINCAD	1
CINCLANT	1
CINCLANTABNCF	1
CINCNORAD	1
CIRCUITS	1
CIVIC	1
CIVILIAN	2
CLASS	2
CLOSE	12
CLOSE-AIR-	1
CLOSE-AIR-SUPPORT	4
CLOSE-AIR-SUPPORT-OPERATIONS	1
CLOSED	1
CLOSED-COURSE	1
CLOSURE	1
CLUSTER	4
COMPUTER	2
COBS	1
COCKPIT	1
CODE	2
COLLABORATE	1
COLLECTION	2
COMBAT	36
COMBAT-READY	3
COMBINATION	1
COMBINE	1
COMBINED	3
COME	1
COMING	2
COMMAND	71
COMMAND'S	1
COMMANDER	23
COMMANDER'S	3
COMMANDERS	5
COMMANDS	4
COMMENSURATE	1
COMMISSION	1
COMMITTED	2
COMMON	2
COMMUNICATION	9
COMMUNICATIONS	14
COMPARABLE	1
COMPASS	3
COMPATIBLE	4
COMPETENT	1

COMPETING	1
COMPLEMENT	1
COMPLEMENTS	1
COMPLETE	4
COMPLETENESS	1
COMPLEX	1
COMPLEXITY	1
COMPONENT	13
COMPONENTS	1
COMPOSED	1
COMPOSITE	1
COMPOSITION	1
COMTAC	1
COMUSAFSO	1
CONCEALMENT	1
CONCENTRATE	1
CONCENTRATED	1
CONCEPT	4
CONCEPTS	7
CONCERNED	1
CONCERNING	2
CONCERT	1
CONDITION	2
CONDITIONS	7
CONDUCT	11 —
CONDUCTED	8
CONDUCTING	1
CONDUCTS	2
CONFIGURATION	2
CONFIGURED	2
CONFLICT	1
CONFLICTS	1
CONFUSE	1 —
CONGRESS	1
CONSEQUENTLY	1
CONSIDERABLE	1
CONSIDERATIONS	1
CONSISTENT	1
CONSISTENTLY	1
CONSISTING	1
CONSISTS	2 —
CONSTANT	3
CONSTITUTE	1
CONSTRAINED	1
CONSTRAINING	1
CONSTRAINS	1 —
CONSTRAINTS	6
CONSULTATION	1
CONSUME	1 —
CONSUMED	1
CONTAINS	6 —
CONTINGUOUS	1
CONTINENTAL	2
CONTINGENCIES	1
CONTINGENCY	1

CONTINUE	3 /
CONTINUES	1
CONTINUITY	1
CONTINUOUS	1
CONTRACT	1
CONTRIBUTE	2
CONTRIBUTES	3
CONTROL	82 /
CONTROLLED	4
CONTROLLER	3
CONTROLLERS	2
CONTROLS	2
CONUS	4
CONVENTIONAL	6
CONVERSELY	1
CONVERSION	1
CONVERTING	1
CONVERTS	1 /
COORDINATE	2 /
COORDINATED	1
COORDINATING	2
COORDINATION	6
COPING	1
CORPS	1
CORRECTIVE	1
CORSAIR	1
COST	2
COST-EFFECTIVE	1
COULD	4
COUNTER	1
COUNTERACTION	1
COUNTERAIR	9
COUNTERED	1 /
COUNTERINSURGENCY	1
COUNTERMEASURES	2
COURSES	1
COVER	1 /
COVERAGE	4
COVERED	1
CRC	2
CREEK	1
CREW	8
CREWMEMBERS	1
CREWS	4
CRISIS	1
CRITERIA	1
CRITICAL	8
CROSSING	1
CRP	1
CRUSADER	1
CSAF	1
CURRENCY	1
CURRENT	10
CURRENTLY	3
D	3



DAILY	4
DAMAGE	6
DART	1
DATA	8
DATA-LINKED	1
DATE	1
DAVIS-	1
DAVIS-MONTHAN	1
DAY	29
DAYS	6
DEAL	1
DECENTRALIZED	7
DECENTRALIZING	1
DECEPTION	1
DECISIONS	2
DECISIVE	1
DECREASE	1
DEDICATED	1
DEEP	1
DEFEND	1
DEFENDED	1
DEFENSE	46
DEFENSE-CAPABLE	1
DEFENSES	5
DEFENSIVE	6
DEFICIENCIES	1
DEFINITELY	1
DEGRADE	4
DEGREE	6
DEGREES	1
DELAY	1
DELEGATED	1
DELIVER	1
DELIVERED	2
DELIVERING	1
DELIVERY	13
DELTA	1
DEMAND	2
DEMANDS	1
DEMONSTRATION	3
DEMONSTRATIONS	1
DENY	1
DEPARTURE	1
DEPENDENCY	2
DEPENDING	1
DEplete	1
DEPLETED	1
DEPLOY	3
DEPLOYED	5
DEPLOYMENT	9
DEPLOYMENTS	1
DEPUTY	2
DESCRIBE	3
DESCRIBED	2
DESCRIPTION	1

DESIGN	4
DESIGNATED	4
DESIGNATION	2
DESIGNATOR	4
DESIGNATORS	2
DESIGNED	21
DESIRABLE	1
DESIRED	2
DESPITE	1
DESTROY	6 /
DESTROYED	3
DESTROYING	4
DESTRUCTION	4
DESTRUCTIVE	1
DETACHMENTS	1
DETAIL	1
DETAILED	2
DETAILS	1
DETECT	3 /
DETECTION	5
DETER	2
DETERMINE	3
DETERRENCE	2
DETONATE	1 /
DEVELOP	3
DEVELOPED	6
DEVELOPING	3
DEVELOPMENT	6
DEVELOPMENTAL	1
DEVELOPMENTS	2
DEVELOPS	2 /
DEVICE	1
DEVICES	4
DEW	1
DG	1
DICTATE	1 /
DID	5
DIFFICULTIES	1
DIFFICULTY	1
DIMENSION	1
DIRECT	4 /
DIRECTED	6
DIRECTING	2
DIRECTION	4
DIRECTIVES	2
DIRECTLY	4
DIRECTS	3
DISCRIMINATION	1
DISCUSSED	3
DISCUSSES	1 /
DISCUSSION	2
DISPENSER	4
DISPENSING	1
DISPERSED	1
DISPLAY	2 /

DISPLAYS	2
COMPOSITION	2
DISRUPT	1
DISRUPTING	1
DISRUPTIVE	1
DISSIMILAR	1
DISSIMILIAR	1
DISTANCE	2
DISTANCES	3
DISTANT	1
DISTRIBUTION	3
DIVIDED	1
DIVISION	3
DIVISIONS	2
DME	1
DO	35
DOCTRINE	7
DOCTRINES	3
DOCUMENT	1
DOES	13
DON	1
DON'T	2
DOWN"	1
DRAGONFLY	1
DRAMATIC	1
DRIVERS	2
DRAW	1
DRAW	2
DROPS	1
DUE	1
DUMB	3
DUPLICATION	1
DURANDAL	3
DURING	7
DUSK	4
DUTY	5
E	8
E-3A	2
EACH	23
E	2
E'S	1
EARLIEST	1
EARLY	7
EARLY-LIGHT	1
EARTH'S	1
EASED	1
EC	1
EC-130E	1
EC-130H	1
EC-135K	2
EC-135P	2
ECOS	1
EXPLONS	1
ECONOMIC	1
ECONOMICAL	1

EFFICIENCY	1
ECONOMY	1
ECS	1
EDGE	3
EF-	1
EF-111A	5
EFFECT	2
EFFECTIVE	8
EFFECTIVELY	4
EFFECTIVENESS	8
EFFECTS	4
EFFORT	16
EFFORT-BASIC	1
EFFORTS	1
EIGHT	1
EITHER	8
EJECTION	4
ELECTRICAL	2
ELECTRO-OPTICAL	1
ELECTROMAGNETIC	1
ELECTRONIC	18
ELECTRONICALLY	1
ELEMENT	2
ELEMENTS	7
ELGIN	1
ELITE	1
EMISSIONS	1
EMIT	1
EMITTERS	1
EMPHASIS	1
EMPLOY	3
EMPLOYED	6
EMPLOYING	4
EMPLOYMENT	14
EMPLOYS	3
EN	1
ENABLE	2
ENABLED	1
ENABLES	2
ENCOMPASS	1
ENCOMPASSES	1
ENDURANCE	1
ENDURING	1
ENEMY	45
ENEMY'S	5
ENEMY-ORIENTED	1
ENERGY	2
ENGAGE	2
ENGAGED	4
ENGAGEMENT	2
ENGINE	1
ENGINEERING	1
ENHANCE	3
ENHANCED	1
ENHANCES	1

FINANCING	1
ENOUGH	10
ENSURE	6
ENSURES	2
ENTERING	1
ENTIRE	2
ENTRY	1
ENVELOPES	1
ENVIRONMENT	3
ENVIRONMENTS	1
EO-GUIDED	1
EQUIP	1
EQUIPMENT	14
EQUIPPED	11
EQUIPPING	1
ESCORTING	1
ESPECIALLY	2
ESSENTIAL	6
ESTABLISH	1
ESTABLISHED	2
ESTABLISHES	1
ESTABLISHING	1
ESTABLISHMENT	2
ESTIMATED	1
ETIC	1
ETIM	2
EVACUATION	2
EVALUATING	1
EVALUATION	10
EVASIVE	1
EVEN	1
EVENING	1
EVENT	1
EVERY	1
EVERYTHING	2
EWO	1
EXAMINES	1
EXCELLENT	1
EXCEPT	1
EXCEPTIONALLY	1
EXCESS	6
EXCHANGED	1
EXECUTE	1
EXECUTED	4
EXECUTES	1
EXECUTING	1
EXECUTION	8
EXERCISE	2
EXERCISES	3
EXERTED	1
EXHIBIT	1
EMERGENCIES	1
EXIST	1
EXISTING	2
EXPECTED	2

EXPENDED	1
EXPERT	1
EXPERTISE	3
EXPLODES	1
EXPLOIT	3
EXPLOITATION	1
EXPLOSIVE	2
EXTEND	1
EXTENDED	2
EXTENSION	1
EXTERNAL	4
EXTRA	2
EXTRACTION	1
EXTRAORDINARY	2
EYE	1
EYE-LEVEL	1
EYES	1
F	2
F-	4
F-105	4
F-106	4
F-111	5
F-111'S	1
F-111F	2
F-111G	13
F-111'S	1
F-16	10
F-16A	1
F-4	5
F-4C	4
F-4D	3
F-4E	10
F-4ES	1
F-4G	6
F-5	1
F-8	1
F5E	1
FAC	2
FACETS	3
FACILITATE	1
FACILITIES	4
FACILITY	1
FACS	1
FACTOR	1
FACTORS	7
FAILS	2
FALCON	4
FAMILY	1
FAR	2
FART	1
FATURABLE	1
FEASIBLE	1
FEATURES	1
FEET	6
FERRY	4

ROAD	3
FIGHT	2
FIGHTER	26
FIGHTERS	4
FIGHTING	3
FIGTHER	1
FILL	2
FILLED	1
FILLSTANDS	1
FINAL	1
FIRE	7
FIRE	1
FEPOWER	4
FIRST	12
FISSS	1
FITTED	3
FIVE	5
FIXED	1
FLAG	2
FLAK	1
FLARES	1
FLEXIBILITY	8
FLIGHT	6
FLIR	2
FLORIDA	2
FLY	4
FLOWN	2
FLY	10
FLY-BY-WIRE	1
FLYING	13
FMC	4
FO	2
FOCAL	2
FOLLOW-ON	2
FOLLOWED	1
FOLLOWING	5
FOLS	1
FOOT	1
FOR	160
FORCE	66
FORCE'S	1
FORCE-DEVELOPED	1
FORCES	103
FORCES'	1
FORCES:	1
FOREIGN	1
FORKLIFTS	1
FORM	8
FORMATION	1
FORMERLY	1
FORWARD	13
FOUR	1
FOUND	1
FOUR	9
FRAGMENTATION	2

FORMING	1
FREE	1
FREE-FALL	2
FRENCH	1
FREQUENCY	1
FREQUENTLY	1
FRIENDLY	20
FROM	37
FUEL	20
FUELS	2
FULL	2
FULLY	3
FUNCTION	4
FUNCTIONS	5
FUNDAMENTAL	3
FUNDED	1
FURNISHES	1
FURTHER	1
FURTHERMORE	1
FUSELAGE	1
FUTURE	3
FUZES	1
G	7
GAIN	3
GAINED	1
GAPPING	2
GAPS	1
GATHERING	1
GATOR	1
GBU-15	1
GEAR	2
GEARED	1
GENERAL	7
GENERAL-PURPOSE	4
GENERALLY	2
GENERATE	1
GENERATED	2
GENERATION	7
GENERATIONS	1
GENIE	2
GEOGRAPHY	1
GET	6
GIVE	1
GIVEN	1
GLIDE	1
GLOBAL	1
GOING	2
GRAVITY-FORCE	1
GREAT	1
GREATER	2
GREATEST	2
GREATLY	3
GROUND	32
GROUND-BASED	1
GROUP	5



GROUPS	1
GUAVE	1
GUARD	1
GUIDANCE	8
GUIDED	8
GUN	2
GUNS	2
GUNSIGHT	1
HAHN	2
HAND	1
HANDBOOKS	1
HANDLE	1
HARDWARE	1
HARM	3
HAS	20 /
HAVE	42
HAVING	2
HE	1
HEADQUARTERS	4
HEADS-UP	2
HEAVIER	1
HEAVILY	1
HELD	1 /
HELP	1 /
HF	1
HIGH	15
HIGH-	1
HIGH-EXPLOSIVE	1
HIGH-PERFORMANCE	1
HIGH-RATE	1
HIGH-SPEED	2
HIGH-VELOCITY	1
HIGHER	3
HIGHEST	1
HIGHLY	1
HOLD	1 /
HOME	2
HOSTILE	8
HOT	2
HOUR	2
HOURS	3
HOW	110
HOWARD	1
HOWEVER	4
HUD	2
HYDRANTS	1
HYDRAULICALLY	1
I	28
ICELAND	2
IDENTIFICATION	7
IDENTIFIED	2
IDENTIFY	3 /
IDENTITY	1
IF	26
IFF	1

7  
IIR 1  
ILLINOIS 1  
IMAGED 1  
IMAGERY 1  
IMAGING 5  
IMMEDIATE 5  
IMPACT 4  
IMPACTED 1  
IMPAIR 1  
IMPLEMENT 4  
IMPLIES 1  
IMPORTANCE 1  
IMPROVE 4  
IMPROVED 9  
IMPROVEMENT 2  
IMPROVEMENTS 1  
IMPROVES 1  
IN 169  
IN-COCKPIT 1  
IN-FLIGHT 2  
IN-THEATER 1  
INADVERTENT 1  
INCENDIARY 1  
INCLUDE 15  
INCLUDED 2  
INCLUDES 5  
INCLUDING 5  
INCORPORATE 1  
INCORPORATES 2  
INCREASE 5  
INCREASED 5  
INCREMENTALLY 1  
INDEPENDENT 1  
INDEPENDENTLY 2  
INDICATIONS 2  
INDIVIDUALS 3  
INDUSTRIAL 1  
INFLUENCE 1  
INFORMATION 11  
INFRARED 9  
INFRARED--SEEKING 1  
INHERENT 2  
INHIBIT 1  
INITIALLY 1  
INITIATE 1  
INITIATES 2  
INITIATION 1  
INITIATIVES 1  
INNOVATION 1  
INTERACTIVE 1  
INPUTS 1  
INSEPARABLE 1  
INSERTION 1  
INSIDE 1

DETECTION	1
INSTALLATIONS	2
INSTRUCTION	1
INSTRUCTIONS	1
INSTRUCTOR	1
INSTRUMENT	2
INSTRUMENTATION	1
INSURANCE	1
INTEGRAL	4
INTEGRATE	1
INTEGRATED	8
INTEGRATING	2
INTEGRATION	6
INTELLIGENCE	10
INTENDED	1
INTENT	1
INTER-AMERICAN	1
INTERACT	1
INTERCEPT	3
INTERCEPTION	2
INTERCEPTOR	3
INTERCEPTORS	1
INTERDICTED	1
INTERDICTION	27
INTEREST	1
INTERFACE	2
INTERFERENCE	1
INTERFERING	1
INTERIM	1
INTERNAL	7
INTERNALLY	1
INTEROPERABILITY	2
INTERPRETING	1
INTERROGATION	1
INTERROGATOR	1
INTERSERVICE	1
INTERTHEATER	2
INTO	11
INTRATHEATER	1
INTRODUCED	1
INTRODUCTION	1
INVENTORIES	1
INVENTORY	4
INVOLVE	2
INVOLVES	1
INVULNERABLE	1
IS	239
ISOLATION	1
ISSUING	1
IT	42
ITEMS	1
ITSELF	19
J-75-P-17	1
J-SEAD	1

	2
JAMMING	4
JAPAN	1
JET	1
JOINT	6
JOINTLY	1
JP-4	2
KADENA	1
KEEP	2
KEESLER	1
KEFLAVIK	1
KEY	1
KILL	4
KILLED	1
KIND	2
KINETIC	1
KIT	1
KITS	1
KNOTS	3
KNOW	1
KNOWN	2
L	1
LAND	13
LAND-BASED	1
LAND-FORCE	1
LANDING	2
LANE	2
LANGLEY	3
LANTCOM	1
LARGE	2
LARGE-	1
LARGEST	1
LASER	10
LASER-GUIDED	4
LASER-SEEKING	1
LAST	1
LATE	2
LATER	1
LATEST	1
LATIN	2
LAUNCH	10
LAUNCHED	2
LAUNCHES	1
LEAD	1
LEAD-	1
LEAD-IN	3
LEADERSHIP	1
LEAVE	1
LET	1
LETHAL	1
LETHALITY	1
LEWIS	11
LEVELS	4
LIAISON	3
LIEU	1

LIGHT	5
LIGHTER	1
LIKELY	1
LIMIT	4
LIMITATION	2
LIMITATIONS	1
LIMITED	5
LIMITING	4
LIMITS	1
LINE	5
LINK	1
LIT	6
LOAD	3
LOADED	2
LOADING	8
LOCAL	3
LOCALIZED	1
LOCATE	1
LOCATED	8
LOCATION	5
LOCATIONS	1
LOCKED-ON	1
LOGISTIC	2
LOGISTICAL	1
LOGISTICS	7
LOITER	1
LOITERING	1
LONG	6
LONG-RANGE	2
LONGER	1
LOOK	2
LOOKING	3
LOSING	1
LOSSES	1
LOST	5
LOW	4
LOW-	1
LOW-ALTITUDE	2
LOW-FLYING	1
LOW-LEVEL	2
LOW-SPEED	1
M	4
M-61	3
MA-1	1
MAC	2
MACH	2
MACH-2	2
MACHINE	1
MA	1
MANIFICATION	1
MAIN	1
MAINTAIN	8
MAINTAINED	3

CONTAINING	3
MAINTAINS	2
MAINTENANCE	10
MAJOR	5 /
MAKE	2
MAKES	1
MAKING	1
MAN	4 /
MANAGEMENT	5
MANAGER	3
MANAGES	6 /
MANAGING	1
MANEUVER	1 /
MANEUVERABLE	1
MANEUVERING	1
MANEUVERS	2
MANNED	1
MANUALS	2
MANY	80
MAP-LIKE	1
MAPPING	1
MARINE	2
MARITIME	1
MARK-SERIES	2
MARKING	2
MATERIAL	3
METERS	4
MAVERICK	8
MAXIMUM	8
MAY	26
MB-4	1
MD-3'S	1
MDS	6
ME	2
MEANS	6
MEASURES	3
MEASURING	1
MECHANICAL	2
MECHANISMS	1
MEDIUM-RANGE	1
MEET	16
MEETING	1
MEMBERS	3
MERS	1
METERS	1
MICHIGAN	1
MID-1980'S	2
MID-DAY	1
MILES	7
MILITARY	12
MILLIMETER	5
MIL	3
MINES	2
MINIGUN	1
MINIMIZING	1

MINIMUM	4
MISSILE	11
MISSILE-FIRING	1
MISSILES	18
MISSION	47
MISSIONS	33
MISSISSIPPI	1
MK	7
MK-20	2
MK-82	1
MK-82S	1
MM	1
MOBILITY	2
MOBS	1
MODEL	4
MODELS	1
MODERATELY	1
MODERN	1
MODERNIZATION	1
MODIFICATION	1
MODIFICATIONS	1
MODIFIED	9
MODULE	1
MONITOR	2
MONITORING	1
MONTH	2
MONTHAN	1
MORE	10
MORNING	1
MOST	11
MOUNTED	1
MOVE	1
MOVED	2
MOVEMENT	9
MOVING	1
MR	3
MTIS	1
MUCH	11
MUCH-NEEDED	1
MULTIMISSION	1
MULTIPLE	2
MULTIPURPOSE	1
MULTISENSOR	1
MUNITION	7
MUNITIONS	29
MUST	21
MUTUAL	1
MY	6
NATION	1
NATIONAL	7
IONS	1
OTO	1
NATURE	1
NAUTICAL	5
NAVAL	6

NAVIGATION	7
NAVIGATIONAL	1
NAVY	6
NAVY-DEVELOPED	1
NECESSARY	7
NEED	3
NEEDED	14
NEEDS	2
NEGATION	1
NELLIS	1
NET	1
NETS	2
NEUTRAL	1
NEUTRALIZATION	1
NEUTRALIZE	1
NEUTRALIZED	2
NEUTRALIZING	1
NEVADA	1
NEW	1
NEWLY	2
NEXT	6
NIGHT	11
NINTH	1
NMC	6
NMCB	1
NMCA	2
NO	2
NOMINATED	1
NON-MISSION	2
NONNUCLEAR	1
NOON	1
NORAD	2
NORMALLY	7
NORTH	3
NOT	12
NOTICE	1
NOW	6
NUCLEAR	7
NUMBER	8
NUMBERED	1
O	1
O-2A	4
OA-37	2
OA-37S	1
OAS	1
OBJECTIVE	14
OBJECTIVES	12
OBJECTS	4
OBLIQUE	1
OBSERVATION	2
OBSERVED	1
OVERVER	1
OCCUR	8
OF	392
OFF	4



BASE	1
OFFENSIVE	13
OFFICER	2
OFFICERS	1
OFFICIAL	1
OFTEN	1
OKLAHOMA	1
ON	46
ON-HAND	2
ON-STATION	2
ONBOARD	1
ONCE	2
ONE	15
ONE-BOMB	1
ONE-PASS	1
ONES	1
ONLY	3
OPCON	1
OPERATE	4
OPERATED	2
OPERATES	3
OPERATING	3
OPERATION	13
OPERATIONAL	16
OPERATIONALLY	2
OPERATIONS	89
OPGAN	1
OPORD	2
OPPORTUNITIES	1
OPTICAL	1
OPTIMUM	1
OPTIONS	1
OR	69
ORBIT	2
ORDER	3
ORDNANCE	6
ORGANIC	1
ORGANIZATION	5
ORGANIZATIONAL	2
ORGANIZE	1
ORGANIZED	1
ORGANIZING	1
ORIGIN	1
ORIGINAL	1
OT	1
OTHER	18
OUR	4
OUT	6
OUTCOME	1
OUTLINING	1
OUTSIDE	2
OUTSTANDING	1
OV-	1
OV-10A	2
OVER	10

OVER-THE-HORIZON	1
OVERALL	3
OVERSEAS	3
OXYGEN	1
P	1
PACAF	1
PACE	2
PACKAGED	1
PADDING	1
PANAMA	2
PANORAMIC	1
PARACHUTE-RETARDED	1
PARAGRAPHS	1
PARALLEL	1
PART	6
PARTICIPATE	2
PARTICULARLY	2
PARTIES	1
PARTLY	1
PARTNER	1
PARTS	3
PARTY	2
PASSAGE	1
PASSIVE	4
PATTERN	5
PAVE	8
PAVING	2
PEACE	1
PEACETIME	1
PEAK	1
PENETRATE	1
PENETRATION	2
PENETRATOR	1
PENNSYLVANIA	1
PENNY	2
PEOPLE	2
PEORIA	1
PEP	1
PER	2
PERCENTAGE	2
PERFORM	6
PERFORMANCE	6
PERFORMED	1
PERFORMING	3
PERFORMS	1
PERMIT	8
PERMITS	3
PERMITTING	2
PERSONNEL	17
PERTAING	2
PHANTOM	5
PHANTOMS	1
PHASE	2
PHASES	1
PICTURE	1

AD-A173 699

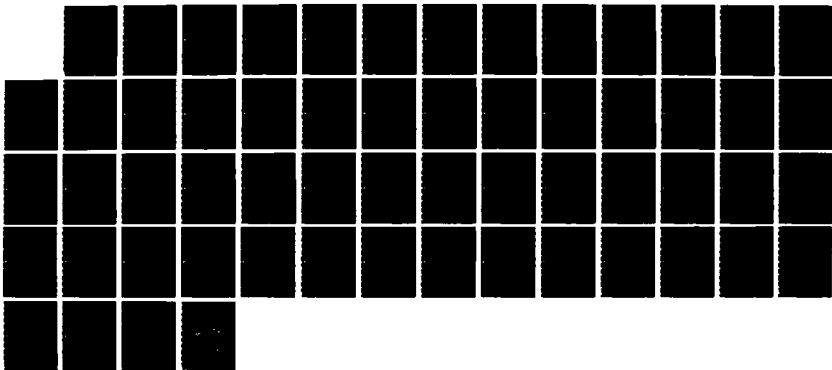
AN EXPERT SYSTEM TO HELP ASSESS TACTICAL AIR READINESS  
AND CAPABILITY PHA (U) SYNERGY INC WASHINGTON DC  
30 MAY 86 F49642-85-D-0035

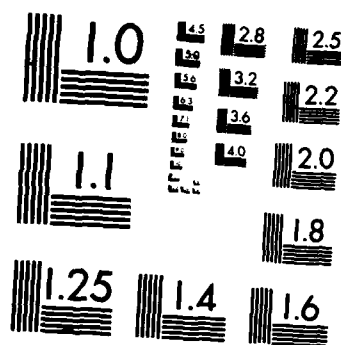
2/2

UNCLASSIFIED

F/G 15/7

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

PILOT	4
PILOT'S	1
PILOTED	1
PILOTS	2
PINPOINT	1
PITS	2
PLACED	2
PLACES	3
PLAN	9
PLANNED	4
PLANNING	5
PLANS	4
PLSS	1
PLUS	2
PMC	1
POD	1
PODS	4
POINT	6
POINTS	1
POL	2
POLICIES	2
POLITICAL	2
POPULATION	2
PORTION	5
PORTIONS	1
POT	1
POTITION	2
POSITIONING	1
POSITIONS	2
POSITIVE	1
POSSESS	1
POSSESSES	1
POSSIBLE	3
POST	3
POSTSTRIKE	1
POSTURE	1
POTENTIAL	2
FOUND	1
POUNDS	8
POWER	1
POWERED	1
PR	1
PRACTICAL	1
PRACTICE	1
PRECISE	1
PRECISION	5
PRECISION-FLYING	1
PRECLUDE	1
PREDETERMINED	1
PREFERRED	1
PREMIUM	1
PREPARE	1
PREPARED	3
PREPLANNED	1
PREREQUISITE	2

PRESCRIBED	1
PRESENT	1
PRESENT-DAY	1
PRESENTED	1
PRESENTLY	1
PRESENTS	2
PRESSURE	1
PREVENT	1
PREVENTING	1
PRIMARY	11
PRINCIPLE	1
PRINCIPLES	2
PRIOR	2
PRIORITIES	2
PRIORITY	4
PROBLEM	4
PROBLEMS	6
PROCEDURAL	1
PROCEDURES	11
PROCESS	5
PROCESS:	1
PROCESSING	4
PROCURED	1
PROCUREMENT	1
PRODUCE	3
PRODUCTION	7
PROFESSIONAL	2
PROGRAM	12
PROGRAMMED	1
PROGRAMS	7
PROGRESS	1
PROJECTED	2
PROJECTS	1
PROPERLY	4
PROPORTION	2
PROPOSED	1
PROTRACTED	1
PROVIDE	28
PROVIDED	9
PROVIDES	26
PROVIDING	6
PROXIMITY	2
PSYCHOLOGICAL	1
PUBLICATIONS	1
PURPOSE	3
PUT	1
PYLON	1
PYLONS	2
QUALIFICATIONS	1
QUALIFIED	6
QUALITATIVE	1
QUARTER	3
QUARTERLY	2
QUICKEST	1
QUICKLY	1

RACK	1
RACK	2
RACKS	2
RADAR	16
RADAR-CONTROLLED	2
RADARS	4
RADIO	1
RANG	1
RANGE	20
RANGING	2
RAPCON	1
RAPID	7
RAPID-	1
RAPIDLY	4
RATE	2
RATHER	2
RATIO	1
RE-ROLE	1
REACH	4
REACTION	2
REACTIVE	1
READ	1
READINESS	4
READY	7
REAL-TIME	1
REALISTIC	4
REALLOCATION	1
REAR	1
REASON	1
RECCE	5
RECEIVE	1
RECEIVED	1
RECEIVER	1
RECEIVERS	1
RECESSED	2
RECLINED	1
RECOGNITION	1
RECOGNIZE	1
RECOMMENDATIONS	1
RECONFIGURATION	1
RECONNAISSANCE	26
RECORD	1
RECORDED	1
RECORDS	3
RECOVER	3
RECOVERED	1
RECOVERY	3
RECUPERATE	1
RED	1
REDEPLOY	1
REDEPLOYMENT	2
REDPLOYMENTS	1
REDUCE	3
REDUCED	2
REDUCTION	1

REFER	1 /
REFERRED	1
REFLECT	1 /
REFLECTED	2
REFUEL	1 /
REFUELING	10
REGARD	1 /
REGARDING	1
REGARDLESS	1
REGIMES	1
REGION	17
RELATED	2
RELATIVE	6
RELATIVELY	2
RELAY	1 /
RELEASE	1
RELEASED	1
RELIABILITY	2
RELIANCE	1
RELIES	1 /
REMAIN	8 /
REPAIR	3 /
REPLACE	3 /
REPORTING	1
REPRESENT	1 /
REPRESENTATIVE	1
REQUEST	1 /
REQUESTS	2
REQUIRE	4 /
REQUIRED	25
REQUIREMENT	8
REQUIREMENTS	16
REQUIRES	4
REQUIRING	2
REQUISITION	1
RESCUE	1 /
RESEARCH	1
RESERVE	4 /
RESERVED	1
RESERVES	1
RESISTANT	1
RESOURCE	3
RESOURCES	22
RESPECT	1 /
RESPECTED	1
RESPECTIVE	1
RESPOND	5 /
RESPONDING	1
RESPONSE	5
RESPONSIBILITIES	4
RESPONSIBILITY	2
RESPONSIBLE	4
RESPONSIVE	3
RESPONSIVENESS	6
REST	1



SEALIFT	2
SEAT	1
SEATING	2
SECOND	2
SECONDS	1
SECURE	4
SECURITY	4
SEE	1
SEEK	2
SEEKER	4
SEEKERS	1
SEEKS	1
SELDOM	1
SELECTED	6
SELECTIVE	1
SELECTIVELY	2
SELF-CONTAINED	4
SELF-PROTECTIVE	1
SEMIACTIVE	2
SENSOR	3
SENSORS	4
SEPARATE	1
SEQUENCE	1
SERIES	4
SEVERELY	2
SERVE	1
SERVES	2
SERVICE	6
SERVICED	1
SERVICES	3
SERVICES*	1
SET	2
SEVEN	1
SHAPED	2
SHARES	1
SHAW	1
SHELTERED	2
SHELTERS	1
SHIFT	2
SHORT	4
SHORTAGE	1
SHORTAGES	1
SHORTFALL	13
SHORTFALLS	6
SHOULD	3
SHOWN	1
SHRIKE	4
SIDE-BY-SIDE	1
SIDE-LOOKING	1
SEE WINDER	6
SQUANT	1
SIGNAL	1
SIGNALING	1
SIGNALS	1

SIGNATURE	1
SIGNIFICANTLY	3
SIMPLE	3
SIMULATE	1
SIMULTANEOUSLY	3
SINCE	2
SINGLE	3
SINGLE-SEAT	1
SINGULAR	1
SIOP	1
SITE	1
SITES	1
SITUATION	2
SITUATIONS	2
SIX	1
SIZE	1
SKILLS	2
SLAR	1
SLC	2
SLOC	1
SLOW	1
SMALL-SCALE	1
SO	5
SOFT	1
SOFTBALL-SIZE	1
SOFTWARE	2
SOEELY	1
SOME	5
SOON	1
SOPHISTICATED	3
SORTIE	12
SORTIES	38
SOURCES	1
SOUTH	1
SOUTHEAST	2
SOUTHERN	4
SOVEREIGNTY	3
SPACE	3
SPANG	1
SPANG'S	1
SPANGDAHLEM	4
SPARE	1
SPARES	5
SPARROW	4
SPARROWS	1
SPECIAL	11
SPECIALISTS	1
SPECIALIZED	2
SPECIALLY	2
SPECIFIC	8
SCIENTIFICALLY	1
SAFIFIED	5
SPECTACULAR	1
SPEED	7

SWORDS	4
SPIKE	2
SPIN	1
SPONSONS	1
SPOT	1
SQUADRON	14
SQUADRONS	6
STAFF	3
STAGE	1
STAND-OFF	1
STANDARD	3
STANDARDS	2
STANDOFF	7
START	3
STARTING	1
STATES	11
STATION	1
STATIONS	2
STATUS	13
STEM	1
STOCKPILE	1
STORED	1
STORES	1
STRAIGHT-WING	1
STRATEGIC	18
STRENGTH	1
STRIKE	5
STRIKES	1
STRUCK	1
STRUCTURE	3
STUDENTS	2
SUBMUNITION	1
SUBMUNITIONS	2
SUBSEQUENT	1
SUBSEQUENTLY	1
SUBSONIC	1
SUBSTITUTE	2
SUBSTITUTION	1
SUBSURFACE	1
SUCCESS	1
SUCCESSFUL	4
SUCH	9
SUFFICIENT	9
SUITED	1
SUMMARY	1
SUPERIORITY	8
SUPERSONIC	2
SUPPLEMENT	1
SUPPLEMENTED	1
SUPPLIES	9
SWIFTLY	3
SUPPORT	91
SUPPORTED	6
SUPPORTING	1
SUPPORTS	1

SUPPRESSING	1
SUPPRESSION	4
SUPREMACY	1
SUPRESSION	1
SURFACE	14
SURFACE-	1
SURFACE-TO-AIR	2
SURFACES	1
SURROUNDINGS	1
SURVEILLANCE	18
SURVIVABILITY	2
SURVIVABLE	1
SUSTAIN	5
SUSTAINABILITY	1
SUSTAINED	6
SUU-11	1
SUU-20	1
SWEEP	1
SWEEPS	1
SWITCHED	1
SYSTEM	46
SYSTEMS	41
T	1
TAC	27
TAC'S	3
TACK	1
TACK-GAINED	2
TACC	5
TACK	4
TACPS	1
TACS	8
TACTICAL	136
TACTICS	12
TAF	2
TAFHQ	2
TAFIG	1
TAIL	1
TAKE	7
TAKEOFF	2
TANDEM	1
TANK	3
TANKERS	1
TANKS	6
TAR	3
TARGET	28
TARGET-ACTIVATED	1
TARGETING	1
TARGETS	23
TASG	3
TASK	8
TASKED	20
TASKING	26
TASKINGS	1
TASKS	8
TASKS:	1

	1
TAXIWAYS	1
TD	1
TDCA	1
TDCS	1
TEACH	1
TEAM	4
TEAMS	1
TECHNIQUES	8
TELEVISION	2
TENETS	1
TEREC	1
TERM	1
TERMS	1
TERRAIN	1
TERRITORY	4
TERS	1
TEST	5
TESTING	1
TEXAS	1
THAN	6
THAT	42
THE	827
THEATER	10
THEATERS	1
THEATRE	1
THEIR	15
THEM	5
THERE	34
THEREBY	1
THERMAL	1
THERMONUCLEAR	1
THESE	30
THEY	15
THEY'LL	1
THIS	49
THOROUGH	1
THOSE	11
THOUSAND-POUND	1
THREAT	9
THREATS	3
THREE	6
THREE-	1
THRESHOLD	1
THRESHOLDS	1
THROUGH	15
THROUGHOUT	2
THRUST	3
THRUST-TO-WEIGHT	1
THUNDERBIRDS	3
THUNDERBOLT	1
THUNDERCHIEF	2
THUS	4
TIME	18
TIME-DELAY	1

TIME-TO-CLIMB	2
TIMELY	5
TIMES	2
TIMING	2
TINKER	2
TIP	1
TIRES	1
TMD	2
TO	320
TO-AIR	2
TO:	2
TODAY	1
TOGETHER	2
TOLERANCE	1
TOMORROW	4
TOPPED	1
TOTAL	4
TOTALLING	1
TOWARD	1
TRACK	1
TRACKER	1
TRACKING	1
TRACT	2
TRACTORS	1
TRADITIONAL	1
TRADOC	2
TRAIN	3
TRAINED	2
TRAINING	32
TRANSCONTINENTAL	1
TRANSFERRED	1
TRANSIT	3
TRANSMITTED	1
TRANSPORT	1
TRANSPORTATION	1
TRAP	2
TRENDS	7
TRIPLE	1
TROOPS	1
TRUCK	1
TRUCKS	5
TURBOJET	1
TURN	2
TURNED	3
TWELFTH	1
TWIN-ENGINE	1
TWO	9
TWO-	1
TWO-MEMBER	1
TYNDALL	1
TYPE	4
TYNS	4
UD-620	1
UHF	2
UNCONVENTIONAL	1

UNDER	11
UNDERSIDE	1
UNDERWING	1
UNEQUIVOCAL	1
UNGUIDED	2
UNIFIED	9
UNIFORM	1
UNILATERAL	1
UNIQUE	1
UNIT	7
UNITED	13
UNITS	21
UNKNOWN	1
UNLIKELY	1
UNLIMITED	2
UNPREPARED	1
UNREFUELED	5
UNRELATED	1
UNTIL	1
UP	6
UPDATES	1
UPDATING	1
UPON	2
US	9
USAF	16
USADWC	1
USAFAGOS	1
USAFE	8
USAFR	2
USAFSO	1
USAF TAWC	1
USAF TFWC	1
USC-5	1
USCINCSO	1
USE	9
USED	5
USEFUL	1
USES	2
USING	5
USREDCOM	1
UTILIZATION	1
UTILIZING	2
UTMOST	1
VALIDATION	1
VALUABLE	2
VARIABLE	1
VARIANT	1
VARIED	1
VARIETY	6
VARIOUS	4
VEHICLE	2
VEHICLES	1
VERSATILE	4
VERSATILITY	1

VISION	2
VHF-AM	2
VHF-FM	2
VIA	3
VIDEO	1
VIEWING	2
VIOLENCE	1
VIRGINIA	1
VIRTUE	1
VISIBILITY	2
VISUAL	3
VISUALLY	1
VITAL	2
VOICE	2
VULCAN	2
VULNERABLE	2
W	1
WAR	7
WARFARE	10
WARGOERS	4
WARHEAD	3
WARHEADS	2
WARNING	17
WARTIME	1
WATER	9
WATER	1
WATER	3
WE	18
WEAPON	19
WEAPONS	56
WEASEL	7
WEASEL'S	1
WEASELS	1
WEATHER	10
WEEK	3
WEeping	2
WEIGH	1
WEIGHT	2
WEIGHTS	1
WELL	2
WERE	4
WHAT	114
WHAT'S	2
WHATS	4
WHEN	16
WHENEVER	1
WHERE	9
WHICH	56
WHICHEVER	1
WHILE	2
WIND	1
WIND	1
WIDE	4
WIDELY	1
WILD	9



W	54
WILLOW	3
WINDOW	2
WINDSHIELD	1
WING	12
WITH	98
WITHIN	15
WITHOUT	6
WORKLOAD	2
WORLD	4
WORLDWIDE	6
WORTH	1
WOULD	1
WSEP	1
WWABNCF	1
X	1
YEAR	1
YEARS	1
YESTERDAY	1
YET	1
YOU	13
ZONE	3
ZONES	2
[NUMBER]	178

vocab.num (by frequency)

TO	827
AND	539
OF	392
TO	320
AIR	279
IS	239
ARE	194
[NUMBER]	178
IN	169
FOR	160
A	155
TACTICAL	136
AIRCRAFT	126
WHAT	114
HOW	110
FORCES	103
BE	100
WITH	98
CAN	93
SUPPORT	91
OPERATIONS	89
CONTROL	82
MANY	80
COMMAND	71
BY	70
ON	69
FORCE	66
AT	58
WEAPONS	56
WHICH	56
AN	54
AS	54
WILL	54
THIS	49
MISSION	47
CAPABILITY	46
DEFENSE	46
ON	46
SYSTEM	46
ENEMY	45
HAVE	42
IT	42
THAT	42
SYSTEMS	41
SORTIES	38
FROM	37
COMBAT	36
DO	35
THERE	34
MISSIONS	33
GROUND	32
TRAINING	32
ALL	30
THESE	30
DAY	29

POSITIONS	29
I	28
PROVIDE	28
TARGET	28
INTERDICTION	27
TAC	27
FIGHTER	26
IF	26
MAY	26
PROVIDES	26
RECONNAISSANCE	26
TASKING	26
ANY	25
CENTER	25
REQUIRED	25
COMMANDER	23
EACH	23
TARGETS	23
AREA	22
RESOURCES	22
AIRBORNE	21
DESIGNED	21
MUST	21
UNITS	21
AGAINST	20
DOUBLE	20
FRIENDLY	20
FUEL	20
HAS	20
RANGE	20
TASKED	20
AIRCROWS	19
AVAILABLE	19
BASE	19
CENTRAL	19
ITS	19
WEAPON	19
ALSO	18
ELECTRONIC	18
MISSILES	18
OTHER	18
SCHEDULE	18
STRATEGIC	18
SURVEILLANCE	18
TIME	18
WE	18
ASSIGNED	17
PERSONNEL	17
REGION	17
WARNING	17
ACFT	16
EFFORT	16
MEET	16
OPERATIONAL	16
RADAR	16

REQUIREMENTS	16
USAF	16
WHEN	16
BASES	15
BOTH	15
HIGH	15
INCLUDE	15
ONE	15
THEIR	15
THEY	15
THROUGH	15
WITHIN	15
ATTACK	14
BATTLE	14
COMMUNICATIONS	14
EMPLOYMENT	14
EQUIPMENT	14
NEEDED	14
OBJECTIVE	14
SQUADRON	14
SURFACE	14
AREAS	13
BOMBS	13
COMPONENT	13
DELIVERY	13
DATE	13
FROM	13
FLYING	13
FORWARD	13
LAND	13
OFFENSIVE	13
OPERATION	13
SHORTFALL	13
STATUS	13
UNITED	13
YOU	13
ATO	12
CAPABILITIES	12
CLOSE	12
FIRST	12
MILITARY	12
NOT	12
OBJECTIVES	12
PROGRAM	12
SORTIE	12
TACTICS	12
WING	12
AIR-TO-AIR	11
AIR-TO-GROUND	11
BEEN	11
CONDUCT	11
EMPLOYED	11
INFORMATION	11
INTO	11
LEVEL	11

MOBILE	11
MOST	11
MUCH	11
NIGHT	11
PRIMARY	11
PROCEDURES	11
SPECIAL	11
STATES	11
THOSE	11
UNDER	11
CARRIED	10
CURRENT	10
ENOUGH	10
EVALUATION	10
F-16	10
F-4E	10
FLY	10
INTELLIGENCE	10
LASER	10
LAUNCH	10
MAINTENANCE	10
MORE	10
OVER	10
REFUELING	10
THEATER	10
WEAPONS	10
WEATHER	10
ACTIVE	9
ASSETS	9
BOMB	9
CENTRALIZED	9
COMMUNICATION	9
COUNTERAIR	9
DEPLOYMENT	9
FOUR	9
IMPROVED	9
INFRARED	9
MODIFIED	9
MOVEMENT	9
PLAN	9
PROVIDED	9
ROLE	9
SUCH	9
SUFFICIENT	9
SUPPLIES	9
THREAT	9
TWO	9
UNIFIED	9
US	9
USE	9
WATER	9
WIRE	9
WILD	9
A-10	8
ACTION	8

AIRAL	8
AEROSPACE	8
AGM-45	8
AIRCREW	8
ARMY	8
AWACS	8
BAI	8
BEING	8
CAMPAIGN	8
CARRY	8
CHIEF	8
CONDUCTED	8
CREW	8
CRITICAL	8
DATA	8
E	8
EFFECTIVE	8
EFFECTIVENESS	8
EITHER	8
EXECUTION	8
FLEXIBILITY	8
FORM	8
GUIDANCE	8
GUIDED	8
HOSTILE	8
INTEGRATED	8
LEAD	8
LOCATED	8
MAINTAIN	8
MAVERICK	8
MAXIMUM	8
NUMBER	8
OCCUR	8
PAVE	8
PERMIT	8
POUNDS	8
REMAIN	8
REQUIREMENT	8
SPECIFIC	8
SUPERIORITY	8
TACS	8
TASK	8
TASKS	8
TECHNIQUES	8
USAFE	8
ACC	7
AFFECT	7
APPROXIMATELY	7
ATTACH	7
BUILT	7
C	7
C	7
CAUSING	7
CONCEPTS	7
CONDITIONS	7

DECENTRALIZED	7
DOCTRINE	7
DURING	7
EARLY	7
ELEMENTS	7
FACTORS	7
FIRE	7
G	7
GENERAL	7
GENERATION	7
IDENTIFICATION	7
II	7
INTERNAL	7
LOGISTICS	7
MILES	7
MK	7
MUNITION	7
NATIONAL	7
NAVIGATION	7
NECESSARY	7
NORMALLY	7
NUCLEAR	7
PRODUCTION	7
PROGRAMS	7
RAPID	7
READY	7
SPEED	7
STANDOFF	7
TAKE	7
TRENDS	7
UNIT	7
WAR	7
WEASEL	7
500-POUND	6
A-7D	6
ABILITY	6
ABLE	6
ACHIEVED	6
ADDITION	6
ANG	6
APPROPRIATE	6
AUTHORITY	6
BATTLEFIELD	6
BEFORE	6
BUT	6
CANNON	6
CHAPTER	6
CONSTRAINTS	6
CONTAINS	6
CONVENTIONAL	6
COORDINATION	6
DAMAGE	6
DAYS	6
DEFENSIVE	6
DEGREE	6



DEVELOP	6
DEVELOPED	6
DEVELOPMENT	6
DIRECTED	6
EMPLOYED	6
ENSURE	6
ESSENTIAL	6
EXCESS	6
F-4G	6
FEET	6
FLIGHT	6
GET	6
INTEGRATION	6
JOINT	6
LINES	6
LONG	6
MANAGES	6
MDS	6
MEANS	6
MY	6
NAVAL	6
NAVY	6
NEXT	6
NMC	6
NOH	6
ORANCE	6
OUT	6
PART	6
PERFORM	6
PERFORMANCE	6
POINT	6
PROBLEMS	6
PROVIDING	6
QUALIFIED	6
RELATIVE	6
RESPONSIVENESS	6
ROCKET	6
SCL	6
SELECTED	6
SERVICE	6
SHORTFALLS	6
SIDEWINDER	6
SQUADRONS	6
SUPPORTED	6
SUSTAINED	6
TANKS	6
THAN	6
THREE	6
UP	6
VARIETY	6
WITHOUT	6
WORLDWIDE	6
ACTIVITIES	5
ADVANCED	5
AGM-78	5

SPACE	5
ANOTHER	5
ARMAMENT	5
AUGMENTATION	5
BASED	5
BASIC	5
BELOW	5
BITBURG	5
BOMBLETS	5
COMMANDERS	5
DEFENSES	5
DEPLOYED	5
DETECTION	5
DID	5
DUTY	5
EF-111A	5
ENEMY'S	5
F-111	5
F-4	5
FIVE	5
FOLLOWING	5
FUNCTIONS	5
GROUP	5
IMAGING	5
IMMEDIATE	5
INCLUDES	5
INCLUDING	5
INCREASE	5
INCREASED	5
LIGHT	5
LIMITED	5
LINE	5
LOCATION	5
LOST	5
MAJOR	5
MANAGEMENT	5
MILLIMETER	5
NAUTICAL	5
ORGANIZATION	5
PATTERN	5
PHANTOM	5
PLANNING	5
PORTION	5
PRECISION	5
PROCESS	5
RECCE	5
RESPOND	5
RESPONSE	5
RUNWAY	5
SO	5
SPARKS	5
SPECIFIED	5
STRIKE	5
SUSTAIN	5

TYPE	5
TEST	5
THEM	5
TIMELY	5
TRUCKS	5
UNREFUELED	5
USED	5
USING	5
000-POUND	4
552D	4
89B	4
ACCURACY	4
ACHIEVE	4
ADDITIONAL	4
ADVERSE	4
AFTER	4
AIM-4F	4
AIR-2A	4
AIRFIELD	4
ALL-WEATHER	4
ALTERNATE	4
ALTITUDE	4
ALTITUDES	4
AVAILABILITY	4
CFM-89	4
CLING	4
CLOSE-AIR-SUPPORT	4
CLUSTER	4
COMMANDS	4
COMPATIBLE	4
COMPLETE	4
CONCEPT	4
CONTROLLED	4
CONUS	4
COULD	4
COVERAGE	4
CREWS	4
DAILY	4
DEGRADE	4
DESIGN	4
DESIGNATED	4
DESIGNATOR	4
DESTROYING	4
DESTRUCTION	4
DEVICES	4
DIRECT	4
DIRECTION	4
DIRECTLY	4
DISPENSER	4
DUSK	4
EFFECTIVELY	4
EFFECTS	4
EJECTION	4
EMPLOYING	4
ENGAGED	4

EXECUTED	4
EXTERNAL	4
F-	4
F-105	4
F-106	4
F-4C	4
FACILITIES	4
FALCON	4
FERRY	4
FIGHTERS	4
FIREPOWER	4
FLOW	4
FMC	4
FUNCTION	4
GENERAL-PURPOSE	4
HEADQUARTERS	4
HOWEVER	4
IMPACT	4
IMPLEMENT	4
IMPROVE	4
INTEGRAL	4
INVENTORY	4
JAMMING	4
KILL	4
LAZER-GUIDED	4
LEVELS	4
LIMIT	4
LIMITING	4
LOW	4
M	4
MAN	4
MATTERS	4
MINIMUM	4
MODEL	4
O-2A	4
OBJECTS	4
OFF	4
OPERATE	4
OUR	4
PASSIVE	4
PILOT	4
PLANNED	4
PLANS	4
PODS	4
PRIORITY	4
PROBLEM	4
PROCESSING	4
PROPERLY	4
RADARS	4
RAPIDLY	4
REACH	4
READINESS	4
REALISTIC	4
REQUIRE	4
REQUIRES	4

NERVE	4
RESPONSIBILITIES	4
RESPONSIBLE	4
RF-4C	4
SEA	4
SECURE	4
SECURITY	4
SEEKER	4
SELF-CONTAINED	4
SENSORS	4
SERIES	4
SHORT	4
SHRIKE	4
SOUTHERN	4
SFANGDAHLEM	4
SPARROW	4
SPEEDS	4
SUCCESSFUL	4
SUPPRESSION	4
TACK	4
TEAM	4
TERRITORY	4
THUS	4
TOMORROW	4
TOTAL	4
TRE	4
TYPES	4
VARIOUS	4
VEHICLES	4
WARGOERS	4
WERE	4
WHATS	4
WIDE	4
WORLD	4
7TH	3
ABCCC	3
ABOVE	3
ACCOMPLISHED	3
ACCURATE	3
ACQUISITION	3
AFTERNOON	3
AGE	3
AGENCY	3
AGGRESSOR	3
AGM-88	3
AIM-7	3
AIM-9	3
ALERT	3
ALLOWS	3
ALTHOUGH	3
AMERICAN	3
ANTI-RADIATION	3
ANYTHING	3
APPROACH	3
ARM	3

A-101	3
ASDC	3
AT-38	3
AVIONICS	3
BASIS	3
BAY	3
BUILD	3
C3	3
CALL	3
CARRIES	3
CATEGORIES	3
CATEGORY	3
CHANGES	3
CHARACTERISTICS	3
COMBAT-READY	3
COMBINED	3
COMMANDER'S	3
COMPASS	3
CONSTANT	3
CONTINUE	3
CONTRIBUTES	3
CONTROLLER	3
CURRENTLY	3
D	3
DEMONSTRATION	3
DEMOBY	3
DESCRIBE	3
DESTROYED	3
DETECT	3
DETERMINE	3
DEVELOP	3
DEVELOPING	3
DIRECTS	3
DISCUSSED	3
DISTANCES	3
DISTRIBUTION	3
DIVISION	3
DOCTRINES	3
DUMB	3
DURANDAL	3
EDGE	3
EMPLOY	3
EMPLOYS	3
ENHANCE	3
ENVIRONMENT	3
EXERCISES	3
EXPERTISE	3
EXPLOIT	3
F-4D	3
FACETS	3
FAD	3
FADING	3
FITTED	3
FULLY	3
FUNDAMENTAL	3

FOUR	3
GAIN	3
GREATLY	3
HARM	3
HIGHER	3
HOURS	3
IDENTIFY	3
INDIVIDUALS	3
INTERCEPT	3
INTERCEPTOR	3
LOTS	3
LANGLEY	3
LEAD-IN	3
LIAISON	3
LINK	3
LOADED	3
LOCAL	3
LOOKING	3
M-61	3
MAINTAINED	3
MAINTAINING	3
MANAGER	3
MATERIAL	3
MEASURES	3
MEMBERS	3
MR	3
NEED	3
NORTH	3
ONLY	3
OPERATES	3
OPERATING	3
ORDER	3
OVERALL	3
OVERSEAS	3
PARTS	3
PERFORMING	3
PERMITS	3
PLACES	3
POSSIBLE	3
POST	3
PREPARED	3
PRODUCE	3
PURPOSE	3
QUARTER	3
RECORDS	3
RECOVER	3
RECOVERY	3
REDUCE	3
REPAIR	3
PLACE	3
RESOURCE	3
RESPONSIVE	3
RESUPPLY	3
ROCKEYE	3

SIZE	3
SCHEDULED	3
SCHOOL	3
SCL'S	3
SENSOR	3
SERVICES	3
SHOULD	3
SIGNIFICANTLY	3
SIMPLE	3
SIMULTANEOUSLY	3
SINGLE	3
SOPHISTICATED	3
SOVEREIGNTY	3
SPACE	3
STAFF	3
STANDARD	3
START	3
STRUCTURE	3
SUPPLY	3
TAC'S	3
TANK	3
TAR	3
TASG	3
THREATS	3
THRUST	3
THUNDERBIRDS	3
TRAIN	3
TRANSIT	3
TURNED	3
VIA	3
VISUAL	3
WARHEAD	3
WAY	3
WEEK	3
WILLOW	3
ZONE	3
0600Z	2
2ND	2
62-MILLIMETER	2
A-7	2
ABOUT	2
ACCESS	2
ACCORDING	2
ACQUIRING	2
ACTING	2
ACTIONS	2
ACTIVITY	2
ADEQUATELY	2
ADVANCE	2
AEROMEDICAL	2
AG	2
AG-2C	2
AGGRESSION	2
AGM-65	2
AGS	2



A-7E	2
A-7F	2
AIR-	2
AIR-LAND	2
AIRCRAFT'S	2
AIRDROP	2
AIRLAND	2
AIRPLANES	2
AIRPOWER	2
ALERTING	2
ALLOCATED	2
ALLOCATION	2
ALREADY	2
AM	2
AMERICA	2
AMPHIBIOUS	2
ANNUAL	2
ANTIARMOR	2
ARMED	2
ARRESTING	2
ASIA	2
ASSEMBLERS	2
ASSEMBLY	2
ASSIGNMENT	2
ASSIST	2
ASSOCIATED	2
AT-58B	2
ATLANTIC	2
ATTAINMENT	2
ATTRITION	2
AUGMENT	2
AVOID	2
AWARE	2
BEHIND	2
BETWEEN	2
BOMBER	2
BOMBERS	2
BOUNDARIES	2
BRIEFED	2
BROAD	2
C2	2
CANISTER	2
CAPACITY	2
CAPS	2
CAREFUL	2
CARRYING	2
CBU-52	2
CBU-58	2
CEM	2
CENTERLINE	2
CHIEF	2
CHANGING	2
CHARACTERISTIC	2
CHARGE	2
CHARGED	2

CIVILIAN	2
CLASS	2
CLUTTER	2
CODE	2
COLLECTION	2
COMING	2
COMMITTED	2
COMMON	2
CONCERNING	2
CONDITION	2
CONDUCTS	2
CONFIGURATION	2
CONFIGURED	2
CONSISTS	2
CONTINENTAL	2
CONTRIBUTE	2
CONTROLLERS	2
CONTROLS	2
COORDINATE	2
COORDINATING	2
COST	2
COUNTERMEASURES	2
CRC	2
DECISIONS	2
DELIVERED	2
DEMAND	2
DEPENDENCY	2
DEPUTY	2
DESCRIBED	2
DESIGNATION	2
DESIGNATORS	2
DESIRED	2
DETAILED	2
DETER	2
DETERRENCE	2
DEVELOPMENTS	2
DEVELOPS	2
DIRECTING	2
DIRECTIVES	2
DISCUSSION	2
DISPLAY	2
DISPLAYS	2
DISPOSITION	2
DISTANCE	2
DIVISIONS	2
DON'T	2
DRIVERS	2
DROP	2
E-3A	2
EAGLE	2
EL-135K	2
EL-135P	2
EFFECT	2
ELECTRICAL	2
ELEMENT	2

ENABLE	2
ENABLES	2
ENERGY	2
ENGAGE	2
ENGAGEMENT	2
ENSURES	2
ENTIRE	2
ESPECIALLY	2
ESTABLISHED	2
ESTABLISHMENT	2
EUCOM	2
EVACUATION	2
EVERYTHING	2
EXERCISE	2
EXISTING	2
EXPECTED	2
EXPLOSIVE	2
EXTENDED	2
EXTRA	2
EXTRAORDINARY	2
F	2
F-111F	2
FAC	2
FAILS	2
FAR	2
FIGHT	2
FILL	2
FLAG	2
FLIR	2
FLORIDA	2
FLOWN	2
FO	2
FOCAL	2
FOLLOW-ON	2
FRAGMENTATION	2
FREE-FALL	2
FUELS	2
FULL	2
GAINING	2
GEAR	2
GENERALLY	2
GENERATED	2
GENIE	2
GOING	2
GREATER	2
GREATEST	2
GUN	2
GUNS	2
HAHN	2
HAVING	2
HIPS-UP	2
HIGH-SPEED	2
HOME	2
HOT	2
HOUR	2

ICELAND	2
IDENTIFIED	2
IMPROVEMENT	2
IN-FLIGHT	2
INCLUDED	2
INCORPORATES	2
INDEPENDENTLY	2
INDICATIONS	2
INHERENT	2
INITIATES	2
INSTALLATIONS	2
INSTRUMENT	2
INTEGRATING	2
INTERCEPTION	2
INTERFACE	2
INTEROPERABILITY	2
INTERTHEATER	2
INVOLVE	2
JAM	2
JP-4	2
KEEP	2
KIND	2
KNOWN	2
LANDING	2
LARGE	2
LATE	2
LATIN	2
LAUNCHED	2
LIMITATION	2
LIT	2
LOGISTIC	2
LONG-RANGE	2
LOOK	2
LOW-ALTITUDE	2
LOW-LEVEL	2
MAC	2
MACH	2
MACH-2	2
MAINTAINS	2
MAKE	2
MANEUVERS	2
MANUALS	2
MARINE	2
MARK-SERIES	2
MARKING	2
ME	2
MECHANICAL	2
MID-1980'S	2
MILES	2
MID-10	2
MOBILITY	2
MONITOR	2
MONTH	2

NOTED	2
MULTIPLE	2
NEEDS	2
NETS	2
NEUTRALIZED	2
NEWLY	2
NMCS	2
NO	2
NON-MISSION	2
NORAD	2
OA-37	2
OBSERVATION	2
OFFICER	2
ON-HAND	2
ON-STATION	2
ONCE	2
OPERATED	2
OPERATIONALLY	2
OPORD	2
ORBIT	2
ORGANIZATIONAL	2
OUTSIDE	2
OV-10A	2
PACE	2
PANAMA	2
PARTICIPATE	2
PARTICULARLY	2
PARTY	2
PAYLOAD	2
PENETRATION	2
PENNY	2
PEOPLE	2
PER	2
PERCENTAGE	2
PERMITTING	2
PERTAINING	2
PHASE	2
PILOTS	2
PITS	2
PLACED	2
PLUS	2
POL	2
POLICIES	2
POLITICAL	2
POPULATION	2
POSITION	2
POSITIONS	2
POTENTIAL	2
PREREQUISITE	2
PRESENTS	2
PRINCIPLES	2
PRIORITY	2
PRIORITIES	2
PROFESSIONAL	2
PROJECTED	2

vocab.num

04-30-1986 13:38:02 PAGE 20

PROPORTION	2
PROXIMITY	2
PYLONS	2
QUARTERLY	2
RACK	2
RACKS	2
RADAR-CONTROLLED	2
RANGING	2
RATE	2
RATHER	2
REACTION	2
RECESSED	2
REDEPLOYMENT	2
REDUCED	2
REFLECTED	2
RELATED	2
RELATIVELY	2
RELIABILITY	2
REQUESTS	2
REQUIRING	2
RESPONSIBILITY	2
RETURNED	2
RF-4CS	2
ROUNDS	2
ROUTING	2
SALIENT	2
SATELLITE	2
SATES	2
SATISFACTORY	2
SATISFIED	2
SCALE	2
SCHEDULES	2
SEAD	2
SEAT	2
SEATING	2
SECOND	2
SEEK	2
SELECTIVELY	2
SEMIACTIVE	2
SERIOUSLY	2
SERVES	2
SET	2
SHAPED	2
SHELTERED	2
SHIFT	2
SINCE	2
SITUATION	2
SITUATIONS	2
SKILLS	2
SOFTWARE	2
SOUTHEAST	2
SPECIALIZED	2
SPECIALLY	2

vocab.num

04-30-1986 13:38:24 PAGE 21

SNAKE	2
STANDARDS	2
STATIONS	2
STUDENTS	2
SUBMUNITIONS	2
SUBSTITUTE	2
SUPERSONIC	2
SURFACE-TO-AIR	2
SURVIVABILITY	2
TAC-GAINED	2
TAF	2
TAFHQ	2
TAKEOFF	2
TELEVISION	2
THROUGHOUT	2
THUNDERCHIEF	2
TIME-TO-CLIMB	2
TIMES	2
TIMING	2
TINKER	2
TMD	2
TO-AIR	2
TO:	2
TOGETHER	2
TRACT	2
TTC	2
TRAINED	2
TRAP	2
TURN	2
UHF	2
UNGUIDED	2
UNLIMITED	2
UPON	2
USAFR	2
USES	2
UTILIZING	2
VALUABLE	2
VARY	2
VERSION	2
VHF-AM	2
VHF-FM	2
VIEWING	2
VISIBILITY	2
VITAL	2
VOICE	2
VULCAN	2
VULNERABLE	2
WARHEADS	2
WEEPING	2
WEIGHT	2
WHAT'S	2
WHILE	2
WINDOW	2
WORKLOAD	2

ES	2
"A"	1
"B"	1
"D"	1
"ILLUMINATION"	1
"LOOK	1
'TILL	1
-	1
000-FEET	1
000-POUND-CLASS	1
10-TON	1
10A	1
111A	1
150-	1
1960S	1
1970S	1
1FS241	1
20-	1
20-MILLIMETER	1
230-GALLON	1
24-HOUR	1
24TH	1
30-	1
30-MILLIMETER	1
370-GALLON	1
400-	1
440-POUND	1
4G	1
6-INCH	1
600-GALLON	1
6ACCS	1
6TH	1
707-320B	1
7D	1
800-POUND	1
82'S	1
844-POUND	1
87B	1
8TH	1
960TH	1
961ST	1
97S	1
9E	1
A-	1
A-7K	1
A-A	1
A-G	1
A10-A	1
A10A	1
AAA	1
AD VENCE	1
AD VOLUTE	1
ACADEMIC	1
ACADEMY	1
ACCEPTABLE	1



ACCOMMODATE	1
ACCOMPANY	1
ACCOMPANYING	1
ACCOMPLISH	1
ACCOMPLISHES	1
ACCOMPLISHING	1
ACCOMPLISHMENT	1
ACCORDED	1
ACCS	1
ACCS'	1
ACHIEVING	1
ACQUIRE	1
ACT	1
ACTUATE	1
ADAPTATION	1
ADAPTED	1
ADDED	1
ADJACENT	1
ADJUST	1
ADJUSTMENT	1
ADTAC	1
ADVANTAGE	1
ADVERSE-WEATHER	1
ADVISE	1
ADVISING	1
AFFECTED	1
AFFECTING	1
AFFORDED	1
AFLC	1
AFSC	1
AGENCIES	1
AGGRESSIVE	1
AGM-65A	1
AGM-65D	1
AGM-65S	1
AHLHORN	1
AI	1
AID	1
AIDS	1
AIM	1
AIM-4G	1
AIM-9J	1
AIM-9L	1
AIM-9P	1
AIR-BREATHING	1
AIR-TO-	1
AIRCRAFT-REMOTELY	1
AIRFRAME	1
AIRLANDING	1
AIRPLANE	1
AIR-TO-STRIKE	1
ALFA	1
ALL-	1
ALLOW	1

ADVISED	1
ALOC	1
ALONE	1
ALTERNATES	1
AMOUNT	1
AMRAAM	1
ANALYSIS	1
ANGLE	1
ANS	1
ANSWER	1
ANTI PERSONNEL	1
ANTI SUBMARINE	1
ANTI TANK	1
ANYWHERE	1
AP	1
APPARENT	1
APPLICABLE	1
APPLICATIONS	1
APPLIED	1
APPORTIONMENT	1
APPROVED	1
APPROXIMATE	1
APR-38	1
ARES	1
ARIZONA	1
ARMOR	1
ARMORED	1
ARMS	1
ARRANGEMENTS	1
ARRIVAL	1
ARRIVES	1
ARSENAL	1
ASPECTS	1
ASRT	1
ASSAULTS	1
ASSESS	1
ASSESSMENT	1
ASSET	1
ASSISTANCE	1
ASSISTS	1
ASSUME	1
ASSURE	1
ASSURED	1
ASSURES	1
ATC	1
ATD	1
ATTACHED	1
ATTACHMENT	1
ATTACKED	1
ATTACKS	1
ATTEMPT	1
ATTEMPTING	1
ATTENTION	1
AUGMENTED	1
AUGMENTING	1

ADJUTANTS	1
AUTHORITATIVE	1
AUTHORITIES	1
AUTHORIZE	1
AUTHORIZED	1
AUTOPILOTS	1
AUXILIARY	1
AWAITING	1
B	1
BACK	1
BASEBALL-SIZE	1
BCE	1
BEACON	1
BEAR	1
BECAUSE	1
BEGAN	1
BEGINS	1
BEHAVIOR	1
BENEFICIAL	1
BERGSTROM	1
BEST	1
BETTER	1
BEYOND	1
BIRD	1
BLU-97	1
BO	1
BOATS	1
BOBTAIL	1
BOOK	1
BOOSTING	1
BORDER	1
BOWING	1
BRANCH	1
BRAND	1
BREAKAGE	1
BREAKTHROUGH	1
BRIEFLY	1
BROUGHT	1
BRUS	1
BU	1
BUILDING	1
C	1
C-130	1
C-135	1
C3I	1
CAMERA	1
CAMERAS	1
CAMOUFLAGE	1
CAN'T	1
CANNONS	1
CAPNOT	1
CALCULATE	1
CAREFULLY	1
CARGO	1
CAROLINA	1

CARRIER	1
CAUSE	1
CBU	1
CBU-87	1
CBU-87B	1
CENTERS	1
CENTRALLY	1
CERTAIN	1
CHALLENGE	1
CHANGE	1
CHANGED	1
CHAPTERS	1
CHOICE	1
CHOOSE	1
CINCAD	1
CINCLANT	1
CINCLANTABNCF	1
CINCNO RAD	1
CIRCUITS	1
CIVIC	1
CLOSE-AIR-	1
	1
CLOSED	1
CLOSED-COURSE	1
CLOSURE	1
C	1
COCKPIT	1
COLLABORATE	1
COMBINATION	1
COMBINE	1
COME	1
COMMAND'S	1
COMMENSURATE	1
COMMISSION	1
COMPARABLE	1
COMPETENT	1
COMPETING	1
COMPLEMENT	1
COMPLEMENTS	1
COMPLETENESS	1
COMPLEX	1
COMPLEXITY	1
COMPONENTS	1
COMPOSED	1
COMPOSITE	1
COMPOSITION	1
COMTAC	1
COMUSAFSO	1
CONCEALMENT	1
CONCENTRATE	1
CONCENTRATED	1
CONCERNED	1
CONCERT	1
CONDUCTING	1
CONFLICT	1

CONFLICTS	1
CONFUSE	1
CONGRESS	1
CONSEQUENTLY	1
CONSIDERABLE	1
CONSIDERATIONS	1
CONSISTENT	1
CONSISTENTLY	1
CONSISTING	1
CONSTITUTE	1
CONSTRAINED	1
CONSTRAINING	1
CONSTRAINS	1
CONSULTATION	1
CONSUME	1
CONSUMED	1
CONTIGUOUS	1
CONTINGENCIES	1
CONTINGENCY	1
CONTINUES	1
CONTINUITY	1
CONTINUOUS	1
CONTRACT	1
CONVERSELY	1
CONVERSION	1
CONVERTING	1
CONVERTS	1
COORDINATED	1
COPING	1
CORPS	1
CORRECTIVE	1
CORSAIR	1
COST-EFFECTIVE	1
COUNTER	1
COUNTERACTION	1
COUNTERED	1
COUNTERINSURGENCY	1
COURSES	1
COVER	1
COVERED	1
CREEK	1
CREWMEMBERS	1
CRISIS	1
CRITERIA	1
CROSSING	1
CRP	1
CRUSADER	1
CSAF	1
CURRENCY	1
DART	1
DATA-LINKED	1
DATA	1
DAVIS-	1
DAVIS-MONTHAN	1
DEAL	1

DECENTRALIZING	1
DECEPTION	1
DECISIVE	1
DECREASE	1
DEDICATED	1
DEEP	1
DEFEND	1
DEFENDED	1
DEFENSE-CAPABLE	1
DEFICIENCIES	1
DEFINITELY	1
DEGREES	1
DELAY	1
DELEGATED	1
DELIVER	1
DELIVERING	1
DELTA	1
DEMANDS	1
DEMONSTRATIONS	1
DENY	1
DEPARTURE	1
DEPENDING	1
DEplete	1
DEPLETED	1
DEPLOYMENTS	1
DESCRIPTION	1
DESIRABLE	1
DESPITE	1
DESTRUCTIVE	1
DETACHMENTS	1
DETAIL	1
DETAILS	1
DETONATE	1
DEVELOPMENTAL	1
DEVICE	1
DEW	1
DG	1
DICTATE	1
DIFFICULTIES	1
DIFFICULTY	1
DIMENSION	1
DISCRIMINATION	1
DISCUSSES	1
DISPENSING	1
DISPERSED	1
DISRUPT	1
DISRUPTING	1
DISRUPTIVE	1
DISSIMILAR	1
DISSIMILIAR	1
DISSIDENT	1
DIVIDED	1
DME	1
DOCUMENT	1
DON	1

DEATH	1
DRAGONFLY	1
DRAMATIC	1
DRONE	1
DROPS	1
DUE	1
DUPLICATION	1
EAGLE'S	1
EARLIEST	1
EARLY-LIGHT	1
EARTH'S	1
EASED	1
EC	1
EC-130E	1
EC-130H	1
EC130S	1
ECHELONS	1
ECONOMIC	1
ECONOMICAL	1
ECONOMIZING	1
ECONOMY	1
ECS	1
EF-	1
EFFORT-BASIC	1
EFFORTS	1
EIGHT	1
ELECTRO-OPTICAL	1
ELECTROMAGNETIC	1
ELECTRONICALLY	1
ELGIN	1
EM	1
EMISSIONS	1
EMIT	1
EMITTERS	1
EMPHASIS	1
EN	1
ENABLED	1
ENCOMPASS	1
ENCOMPASSES	1
ENDURANCE	1
ENDURING	1
ENEMY-ORIENTED	1
ENGINE	1
ENGINEERING	1
ENHANCED	1
ENHANCES	1
ENHANCING	1
ENTERING	1
ENTRY	1
ENVELOPES	1
ENVIRONMENTS	1
ENVOYED	1
EQUIP	1
EQUIPPING	1
ESCORTING	1

ESTABLISH	1
ESTABLISHES	1
ESTABLISHING	1
ESTIMATED	1
ETIC	1
EVALUATING	1
EVASIVE	1
EVEN	1
EVENING	1
EVENT	1
EVERY	1
EWO	1
EXAMINES	1
EXCELLENT	1
EXCEPT	1
EXCEPTIONALLY	1
EXCHANGED	1
EXECUTE	1
EXECUTES	1
EXECUTING	1
EXERTED	1
EXHIBIT	1
EXIGENCIES	1
EXIST	1
EXTENDED	1
EXHAUST	1
EXPLODES	1
EXPLOITATION	1
EXTEND	1
EXTENSION	1
EXTRACTION	1
EYE	1
EYE-LEVEL	1
EYES	1
F-111'S	1
F-15'S	1
F-16A	1
F-4ES	1
F-5	1
F-8	1
F5E	1
FACILITATE	1
FACILITY	1
FACS	1
FACTOR	1
FAMILY	1
FAST	1
FAVORABLE	1
FEASIBLE	1
FEATURES	1
FIGHTER	1
FAILED	1
FILLSTANDS	1
FINAL	1
FIRE	1



FUSS	1
FIXED	1
FLAK	1
FLARES	1
FLY-BY-WIRE	1
FOLLOWED	1
FOLS	1
FOOT	1
FORCE'S	1
FORCE-DEVELOPED	1
FORCES'	1
FORCES:	1
FOREIGN	1
FORKLIFTS	1
FORMATION	1
FORMERLY	1
FOT	1
FOUND	1
FRAMING	1
FREE	1
FRENCH	1
FREQUENCY	1
FREQUENTLY	1
FUNDED	1
FISHES	1
FUTHER	1
FURTHERMORE	1
FUSELAGE	1
FUZES	1
GAINED	1
GAPS	1
GATHERING	1
GATOR	1
GBU-15	1
GEARED	1
GENERATE	1
GENERATIONS	1
GEOGRAPHY	1
GIVE	1
GIVEN	1
GLIDE	1
GLOBAL	1
GRAVITY-FORCE	1
GREAT	1
GROUND-BASED	1
GROUPS	1
GROVE	1
GUARD	1
GUNSIGHT	1
HAND	1
HAND BOOKS	1
HANDLE	1
HARDWARE	1
HE	1
HEAVIER	1

HABITILY	1
HELD	1
HELP	1
HF	1
HIGH-	1
HIGH-EXPLOSIVE	1
HIGH-PERFORMANCE	1
HIGH-RATE	1
HIGH-VELOCITY	1
HIGHEST	1
HIGHLY	1
HOLD	1
HOWARD	1
HYDRANTS	1
HYDRAULICALLY	1
IDENTITY	1
IFF	1
IIR	1
ILLINOIS	1
IMAGED	1
IMAGERY	1
IMPACTED	1
IMPAIR	1
IMPLIES	1
IMPORTANCE	1
IMPROVEMENTS	1
IMPROVES	1
IN-COCKPIT	1
IN-THEATER	1
INADVERTENT	1
INCENDIARY	1
INCORPORATE	1
INCREMENTALLY	1
INDEPENDENT	1
INDUSTRIAL	1
INFLUENCE	1
INFRARED-SEEKING	1
INHIBIT	1
INITIALLY	1
INITIATE	1
INITIATION	1
INITIATIVES	1
INNOVATION	1
INOPERATIVE	1
INPUTS	1
INSEPARABLE	1
INSERTION	1
INSIDE	1
INSPECTION	1
INSTRUCTION	1
INSTRUCTIONS	1
INSTRUCTOR	1
INSTRUMENTATION	1
INSURANCE	1
INTEGRATE	1

INTENDED	1
INTENT	1
INTER-AMERICAN	1
INTERACT	1
INTERCEPTORS	1
INTERDICTED	1
INTEREST	1
INTERFERENCE	1
INTERFERING	1
INTERIM	1
INTERNALLY	1
INTERPRETING	1
INTERROGATION	1
INTERROGATOR	1
INTERSERVICE	1
INTRATHEATER	1
INTRODUCED	1
INTRODUCTION	1
INVENTORIES	1
INVOLVES	1
INVULNERABLE	1
ISOLATION	1
ISSUING	1
ITEMS	1
ITSELF	1
J-17	1
J-SEAD	1
JAPAN	1
JET	1
JOINTLY	1
KADENA	1
KEESLER	1
KEFLAVIK	1
KEY	1
KILLED	1
KINETIC	1
KIT	1
KITS	1
KNOW	1
L	1
LAND-BASED	1
LAND-FORCE	1
LANTCOM	1
LARGE-	1
LARGEST	1
LASER-SEEKING	1
LAST	1
LATER	1
LATEST	1
LAUNCHES	1
LEAD-	1
LEADERSHIP	1
LEAVE	1
LET	1

LETHAL	1
LETHALITY	1
LIEU	1
LIGHTER	1
LIKELY	1
LIMITATIONS	1
LIMITS	1
LINE'S	1
LOADING	1
LOCALIZED	1
LOCATE	1
LOCATIONS	1
LOCKED-ON	1
LOGISTICAL	1
LOITER	1
LOITERING	1
LONGER	1
LOSING	1
LOSSES	1
LOW-	1
LOW-FLYING	1
LOW-SPEED	1
MA-1	1
MACHINE	1
MADE	1
MANIFICATION	1
MAIN	1
MAKES	1
MAKING	1
MANAGING	1
MANEUVER	1
MANEUVERABLE	1
MANEUVERING	1
MANNED	1
MAP-LIKE	1
MAPPING	1
MARITIME	1
MB-4	1
MD-3'S	1
MEASURING	1
MECHANISMS	1
MEDIUM-RANGE	1
MEETING	1
MERS	1
METERS	1
MICHIGAN	1
MID-DAY	1
MINIGUN	1
MINIMIZING	1
MISSILE-FIRING	1
MISSISSIPPI	1
MK-482	1
MK-82S	1
MM	1
MOBS	1

MODELS	1
MODERATELY	1
MODERN	1
MODERNIZATION	1
MODIFICATION	1
MODIFICATIONS	1
MODULE	1
MONITORING	1
MONTHAN	1
MORNING	1
MOUNTED	1
MOVE	1
MOVING	1
MTIS	1
MUCH-NEEDED	1
MULTIMISSION	1
MULTIPURPOSE	1
MULTISENSOR	1
MUTUAL	1
NATION	1
NATIONS	1
NATO	1
NATURE	1
NAVIGATIONAL	1
NAVY-DEVELOPED	1
NEGOTIATION	1
NETS	1
NET	1
NEUTRAL	1
NEUTRALIZATION	1
NEUTRALIZE	1
NEUTRALIZING	1
NEVADA	1
NEW	1
NINTH	1
NMCB	1
NOMINATED	1
NONNUCLEAR	1
NOON	1
NOTICE	1
NUMBERED	1
O	1
OA-37S	1
OAS	1
OBLIQUE	1
OBSERVED	1
OBSERVER	1
OFF-BASE	1
OFFICERS	1
OFFICIAL	1
OFTEN	1
OHIOHOMA	1
ONBOARD	1
ONE-BOMB	1
ONE-PASS	1

OPCON	1
OPLAN	1
OPPORTUNITIES	1
OPTICAL	1
OPTIMUM	1
OPTIONS	1
ORGANIC	1
ORGANIZE	1
ORGANIZED	1
ORGANIZING	1
ORIGIN	1
ORIGINAL	1
OT	1
OUTCOME	1
OUTLINING	1
OUTSTANDING	1
OV-	1
OVER-THE-HORIZON	1
OXYGEN	1
P	1
PACAF	1
PACKAGED	1
PADDING	1
PANORAMIC	1
PARACHUTE-RETARDED	1
PARAGRAPHS	1
PARALLEL	1
PARTIES	1
PARTLY	1
PARTNER	1
PASSAGE	1
PEACE	1
PEACETIME	1
PEAK	1
PENETRATE	1
PENETRATOR	1
PENNSYLVANIA	1
PEORIA	1
PEP	1
PERFORMED	1
PERFORMS	1
PHANTOMS	1
PHASES	1
PICTURE	1
PILOT'S	1
PILOTED	1
PINPOINT	1
PLSS	1
PMC	1
POINTS	1
PORTIONS	1
POSE	1
POSITIONING	1

POSITIVE	1
POSSESS	1
POSSESSES	1
POSTSTRIKE	1
POSTURE	1
POUND	1
POWER	1
POWERED	1
PR	1
PRACTICAL	1
PRACTICE	1
PRECISE	1
PRECISION-FLYING	1
PRECLUDE	1
PREDETERMINED	1
PREFERRED	1
PREMIUM	1
PREPARE	1
PREPLANNED	1
PRESCRIBED	1
PRESENT	1
PRESENT-DAY	1
PRESENTED	1
PRESENTLY	1
PRESSURE	1
PREVENT	1
PREVENTING	1
PRINCIPLE	1
PROCEDURAL	1
PROCESS:	1
PROCURED	1
PROCUREMENT	1
PROGRAMMED	1
PROGRESS	1
PROJECTS	1
PROPOSED	1
PROTRACTED	1
PSYCHOLOGICAL	1
PUBLICATIONS	1
PUT	1
PYLON	1
QUALIFICATIONS	1
QUALITATIVE	1
QUICKEST	1
QUICKLY	1
R-14	1
RADIO	1
RANG	1
RAPCON	1
RAPID-	1
RAPID	1
REARABLE	1
REACTIVE	1
READ	1
REAL-TIME	1

LOCATION	1
REAR	1
REASON	1
RECEIVE	1
RECEIVED	1
RECEIVER	1
RECEIVERS	1
RECLINED	1
RECOGNITION	1
RECOGNIZE	1
RECOMMENDATIONS	1
RECONFIGURATION	1
RECORD	1
RECORDED	1
RECOVERED	1
RECUPERATE	1
RED	1
REDEPLOY	1
REDEPLOYMENTS	1
REDUCTION	1
REFER	1
REFERRED	1
REFLECT	1
REFUEL	1
REGARD	1
RECORDING	1
REGARDLESS	1
REGIMES	1
RELAY	1
RELEASE	1
RELEASED	1
RELIANCE	1
RELIES	1
REPORTING	1
REPRESENT	1
REPRESENTATIVE	1
REQUEST	1
REQUISITION	1
RESCUE	1
RESEARCH	1
RESERVED	1
RESERVES	1
RESISTANT	1
RESPECT	1
RESPECTED	1
RESPECTIVE	1
RESPONDING	1
REST	1
RESTRICT	1
RESTRICTED	1
RESULT	1
REAPPLIED	1
RETARDING	1
RETENTION	1
RETURNS	1



ROCKET-BOOSTED	1
ROCKETS	1
ROLES	1
ROTDOME	1
ROUGHLY	1
ROUTE	1
ROUTES	1
RPVS	1
RTU	1
RUGGED	1
RUNNING	1
RUNWAYS	1
RWR	1
S	1
SANCTUARIES	1
SATISFY	1
SCENE	1
SCL-27'S	1
SCLB	1
SCOPE	1
SCRAMBLE	1
SEALIFT	1
SECONDS	1
SEEKERS	1
SEEKS	1
SELDOM	1
SELECTIVE	1
SELF-PROTECTIVE	1
SEPARATE	1
SEQUENCE	1
SERVE	1
SERVICED	1
SERVICES'	1
SEVEN	1
SHARES	1
SHAW	1
SHELTERS	1
SHORTAGE	1
SHORTAGES	1
SHOWN	1
SIDE-BY-SIDE	1
SIDE-LOOKING	1
SIGINT	1
SIGNAL	1
SIGNALING	1
SIGNALS	1
SIGNATURE	1
SIMULATE	1
SINGLE-SEAT	1
SINGULAR	1
SIOP	1
SITE	1

SIXES	1
SIX	1
SIZE	1
SLAR	1
SLOC	1
SLOW	1
SMALL-SCALE	1
SOFT	1
SOFTBALL-SIZE	1
SOLE	1
SOLELY	1
SOON	1
SOURCES	1
SOUTH	1
SPANG	1
SPANG'S	1
SPARE	1
SPARROWS	1
SPECIALISTS	1
SPECIFICALLY	1
SPECTACULAR	1
SPIN	1
SPONSONS	1
SPOT	1
STAGE	1
STAND-OFF	1
STARTING	1
STATION	1
STEM	1
STOCKPILE	1
STORED	1
STORES	1
STRAIGHT-WING	1
STRENGTH	1
STRIKES	1
STRUCK	1
SUBMUNITION	1
SUBSEQUENT	1
SUBSEQUENTLY	1
SUBSONIC	1
SUBSTITUTION	1
SUBSURFACE	1
SUCCESS	1
SUITED	1
SUMMARY	1
SUPPLEMENT	1
SUPPLEMENTED	1
SUPPORTING	1
SUPPORTS	1
SUPPRESSING	1
SUTEMACY	1
SUCCESSION	1
SURFACE-	1
SURFACES	1
SURROUNDINGS	1

SURVIVABLE	1
SUSTAINABILITY	1
SUU-11	1
SUU-20	1
SWEEP	1
SWEEPS	1
SWITCHED	1
T	1
TAC-	1
TACPS	1
TAFIG	1
TAIL	1
TANDEM	1
TANKERS	1
TARGET-ACTIVATED	1
TARGETING	1
TASKINGS	1
TASKS:	1
TASS	1
TAXIWAYS	1
TD	1
TDCA	1
TDCS	1
TEACH	1
TEAMS	1
TENTS	1
TEREC	1
TERM	1
TERMS	1
TERRAIN	1
TERS	1
TESTING	1
TEXAS	1
THEATERS	1
THEATRE	1
THEREBY	1
THERMAL	1
THERMONUCLEAR	1
THEY'LL	1
THOROUGH	1
THOUSAND-POUND	1
THREE-	1
THRESHOLD	1
THRESHOLDS	1
THRUST-TO-WEIGHT	1
THUNDERBOLT	1
TIME-DELAY	1
TIP	1
TIRES	1
TODAY	1
TOLERANCE	1
TOMED	1
TOTALLING	1
TOWARD	1
TRACK	1

TRUCKER	1
TRACKING	1
TRACTORS	1
TRADITIONAL	1
TRANSCONTINENTAL	1
TRANSFERRED	1
TRANSMITTED	1
TRANSPORT	1
TRANSPORTATION	1
TRIPLE	1
TROOPS	1
TRUCK	1
TURBOJET	1
TWELFTH	1
TWIN-ENGINE	1
TWO-	1
TWO-MEMBER	1
TYNDALL	1
UD0620	1
UNCONVENTIONAL	1
UNDERSIDE	1
UNDERWING	1
UNEQUIVOCAL	1
UNIFORM	1
UNILATERAL	1
UNIQUE	1
UNKNOWN	1
UNLIKELY	1
UNPREPARED	1
UNRELATED	1
UNTIL	1
UPDATES	1
UPDATING	1
USAFADWC	1
USAFAGOS	1
USAFSO	1
USAF TAWC	1
USAF TFWC	1
USC-5	1
USCINCSO	1
USEFUL	1
USREDCOM	1
UTILIZATION	1
UTMOST	1
VALIDATION	1
VARIABLE	1
VARIANT	1
VARIED	1
VEHICLE	1
VERSATILE	1
VERSATILITY	1
VIO	1
VIOLENCE	1
VIRGINIA	1
VIRTUE	1

vocab.num

04-30-1986 13:46:36 PAGE 43

WALLY	1
W	1
WARTIME	1
WATER	1
WEASEL'S	1
WEASELS	1
WEIGH	1
WEIGHTS	1
WHENEVER	1
WHICHEVER	1
WHO	1
WHY	1
WIDELY	1
WINDSHIELD	1
WORTH	1
WOULD	1
WSEP	1
WWABNCF	1
X	1
YEAR	1
YEARS	1
YESTERDAY	1
YET	1

END

12-86

DTIC